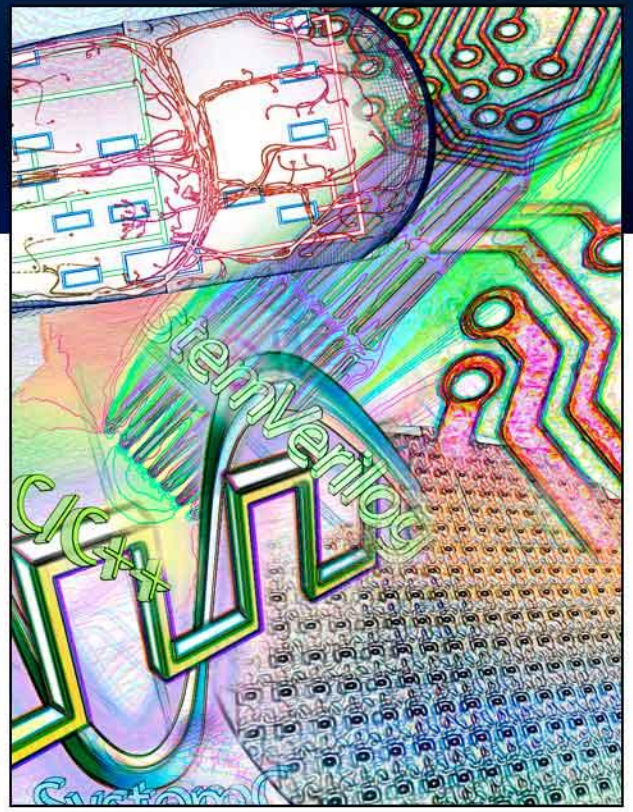


Questa Debug Essentials

Joseph Rodriguez
joe_rodriguez@mentor.com
FPGA Market Development Engineer
Design Verification Division

May 21, 2013

Mentor Graphics®

Agenda: Questa Debug Essentials

- General Questa Operations
- Managing Simulation Messages
- Source Level Debugging
- Viewing and Working with Waveforms
- Root Cause Analysis

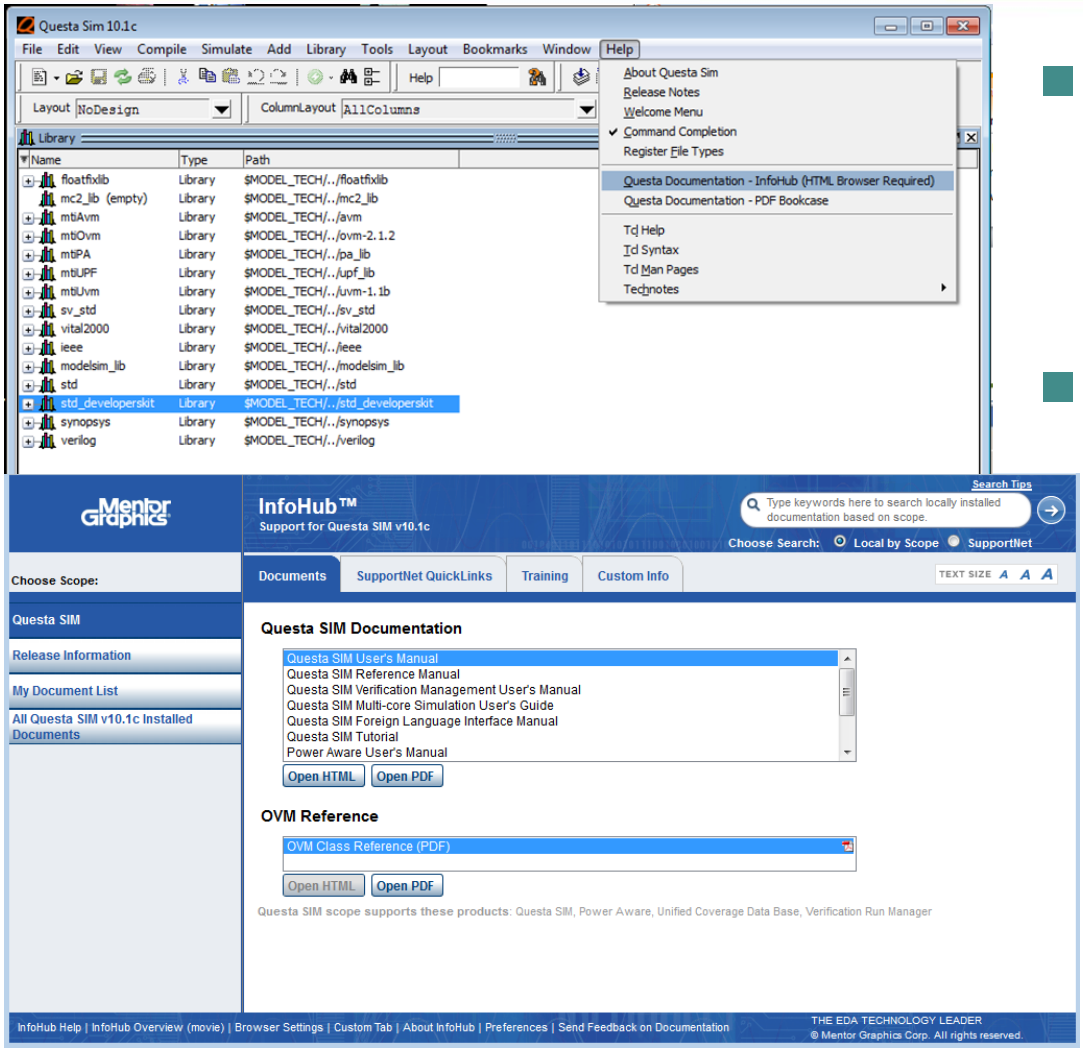
© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Agenda: Questa Debug Essentials

- General Questa Operations
- Managing Simulation Messages
- Source Level Debugging
- Viewing and Working with Waveforms
- Root Cause Analysis

© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

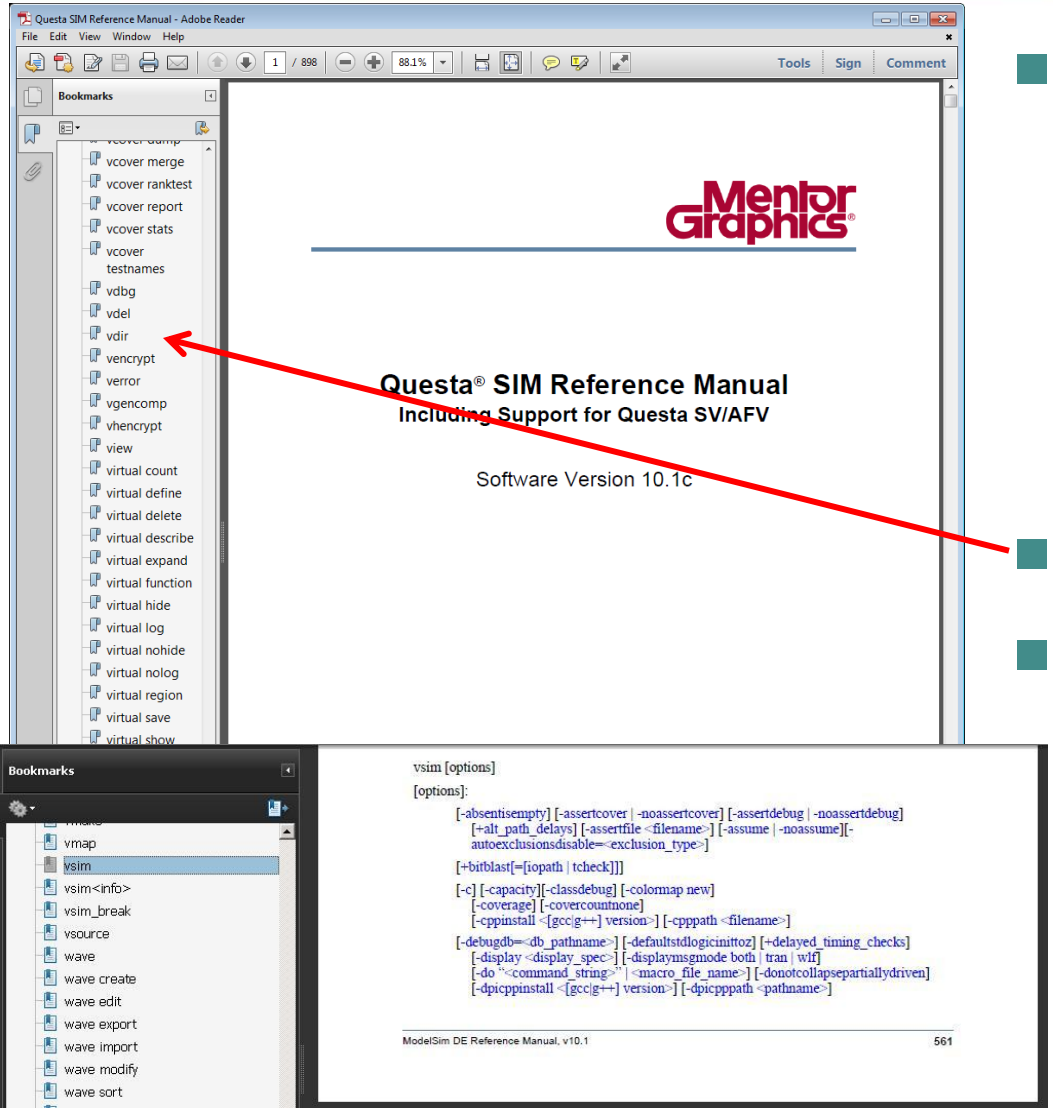
Questa Infohub



- The Infohub is our Simulation Library
  - Access through the tool's "Help" menu
- Access or search all documentation for the tool
  - Release notes
  - User's Manual
  - Command Reference
  - Licensing and Installation

© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

A Reference for Any Operation



- The Command Reference contains a list of every command with every possible option to that command
- Organized by command
- Every command option is explained in detail, some with example usages

© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Questa's Multi-Docking-Interface (MDI)

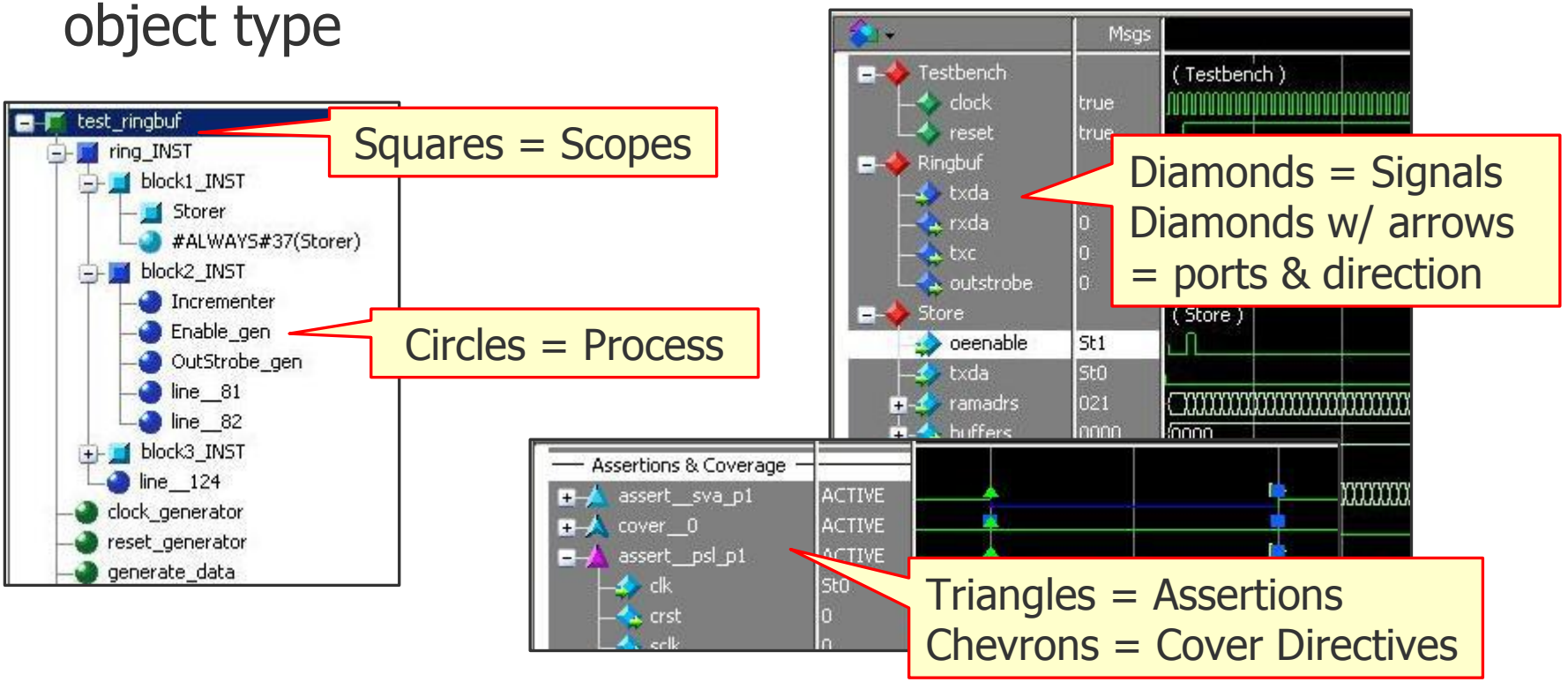
- Questa has many tools for different simulation tasks
  - Each tool has can operate in one or more windows
- Questa's GUI is very configurable allowing for best window placement during each task
- There are several default views for common tasks
  - Create as many custom views as needed
- Any window can be positioned inside Questa's main window frame or as separate window outside the main frame
- Windows can be alone or grouped with other windows in "tab groups"

© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Descriptive Icons

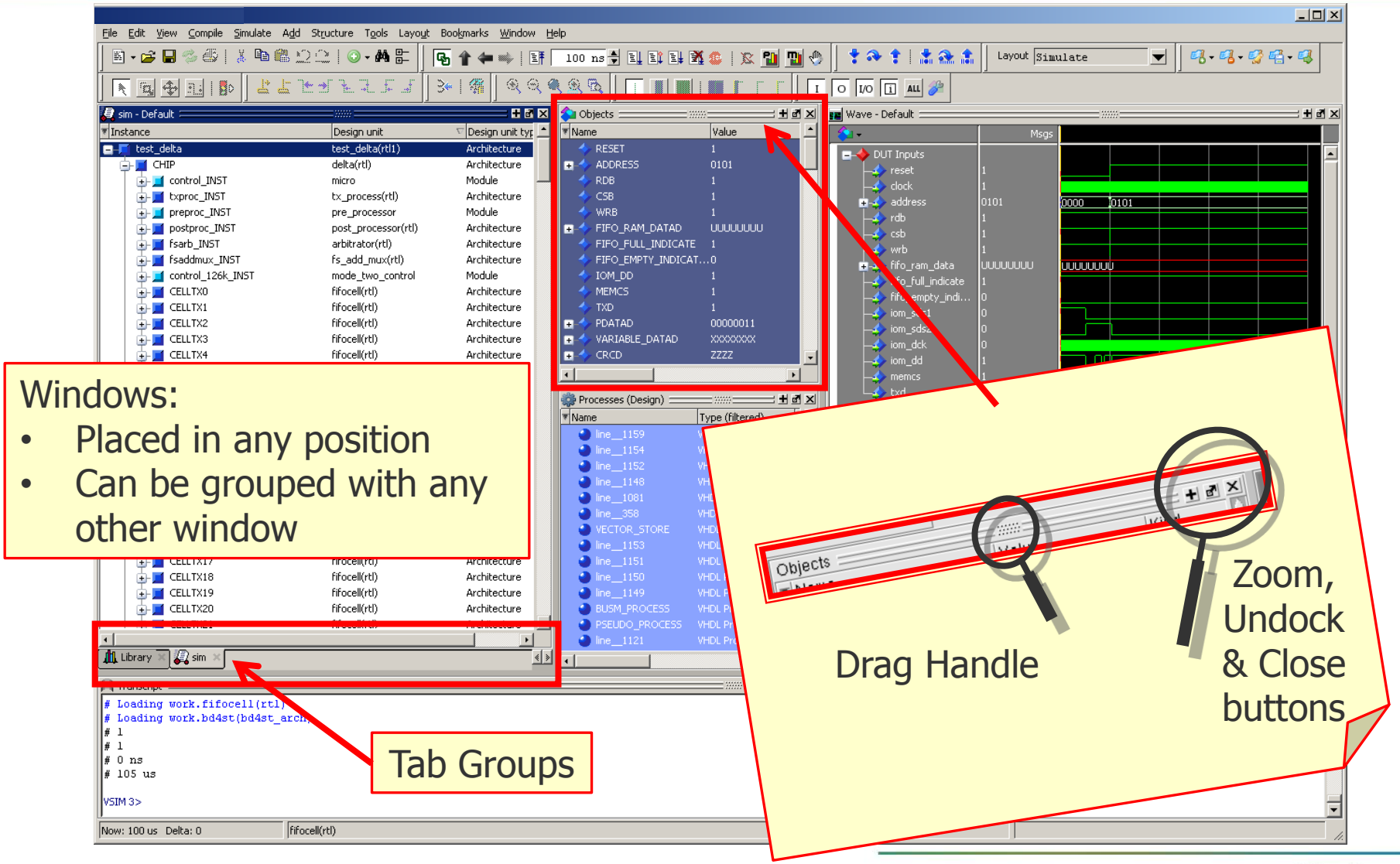
- Different colors represent language
- Different shapes represent object type

Light Blue = Verilog/SystemVerilog
Dark Blue = VHDL
Green = SystemC
Magenta = PSL



© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Basic Window Operations



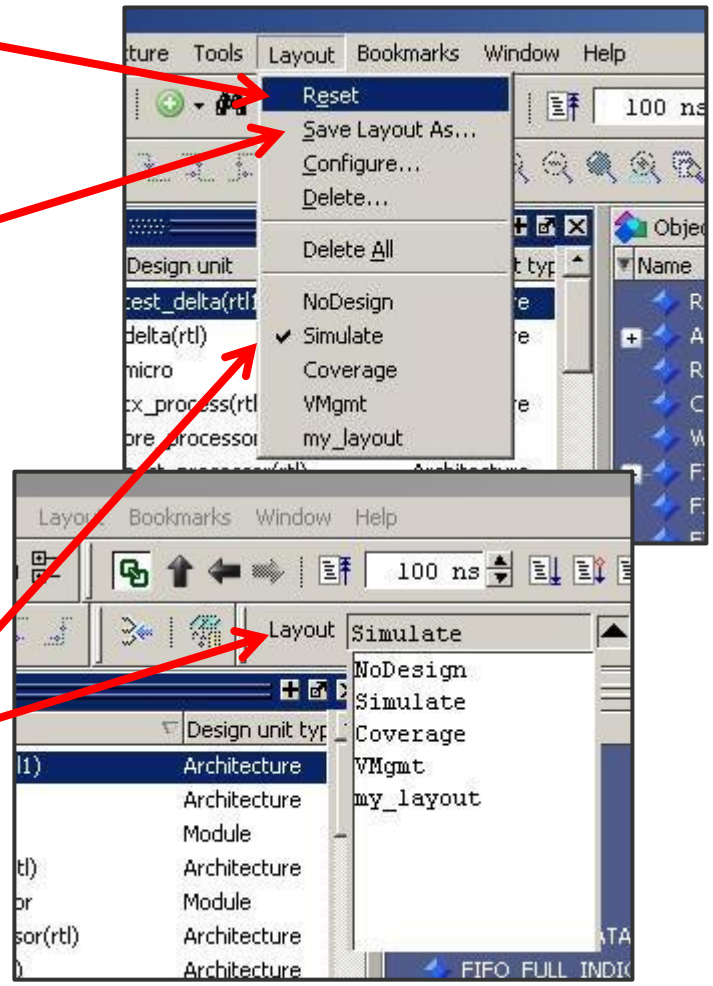
© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Working with Window Layouts

- The *Layout* menu offers the ability to:
  - — *Reset* the tool to the default

Hint: Reset can be very helpful when your windows get "messy"

  - — *Save* any layouts that you like

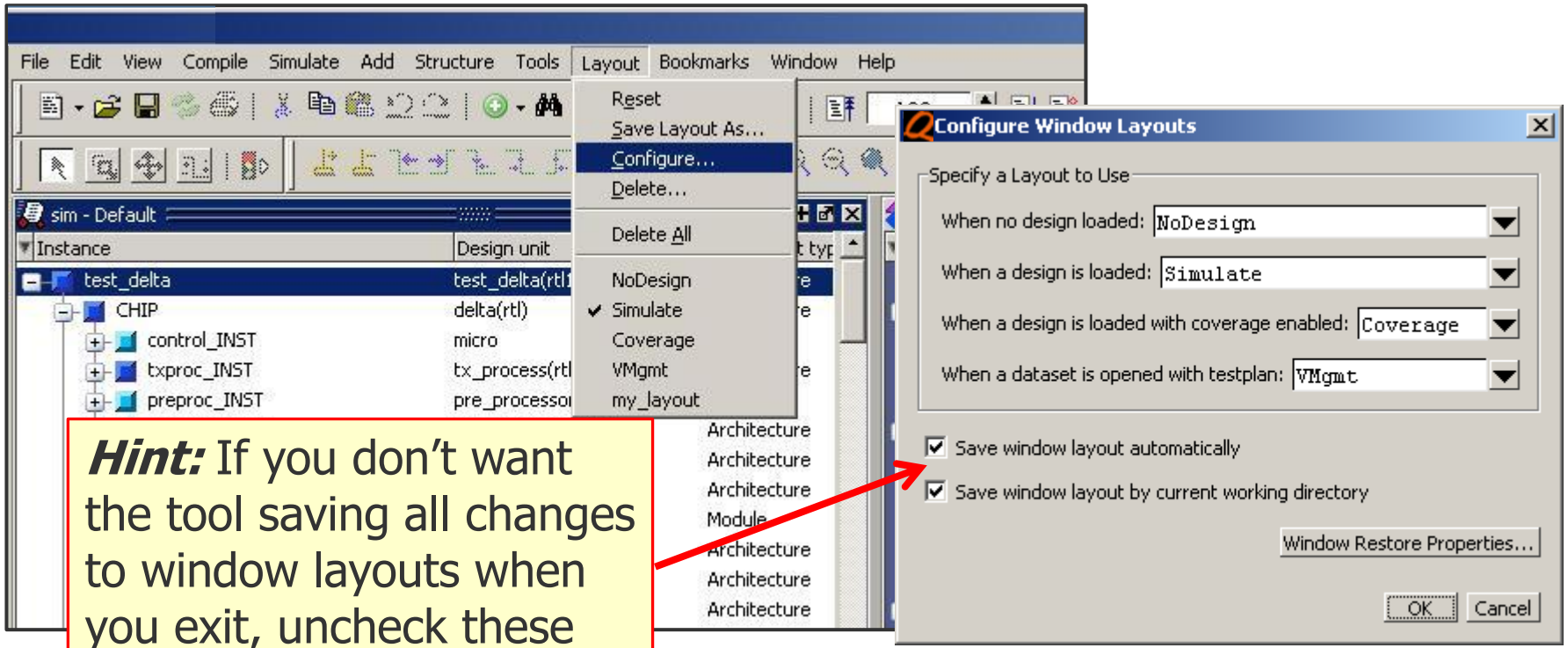


- Saved layouts as well as several predefined layouts are available from the menu or using the *Layout* gadget in the toolbar

© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Layouts for Different Simulation Modes

- Each Simulation mode has a default layout
- Use the Configure dialog to assign your own saved layouts to a mode

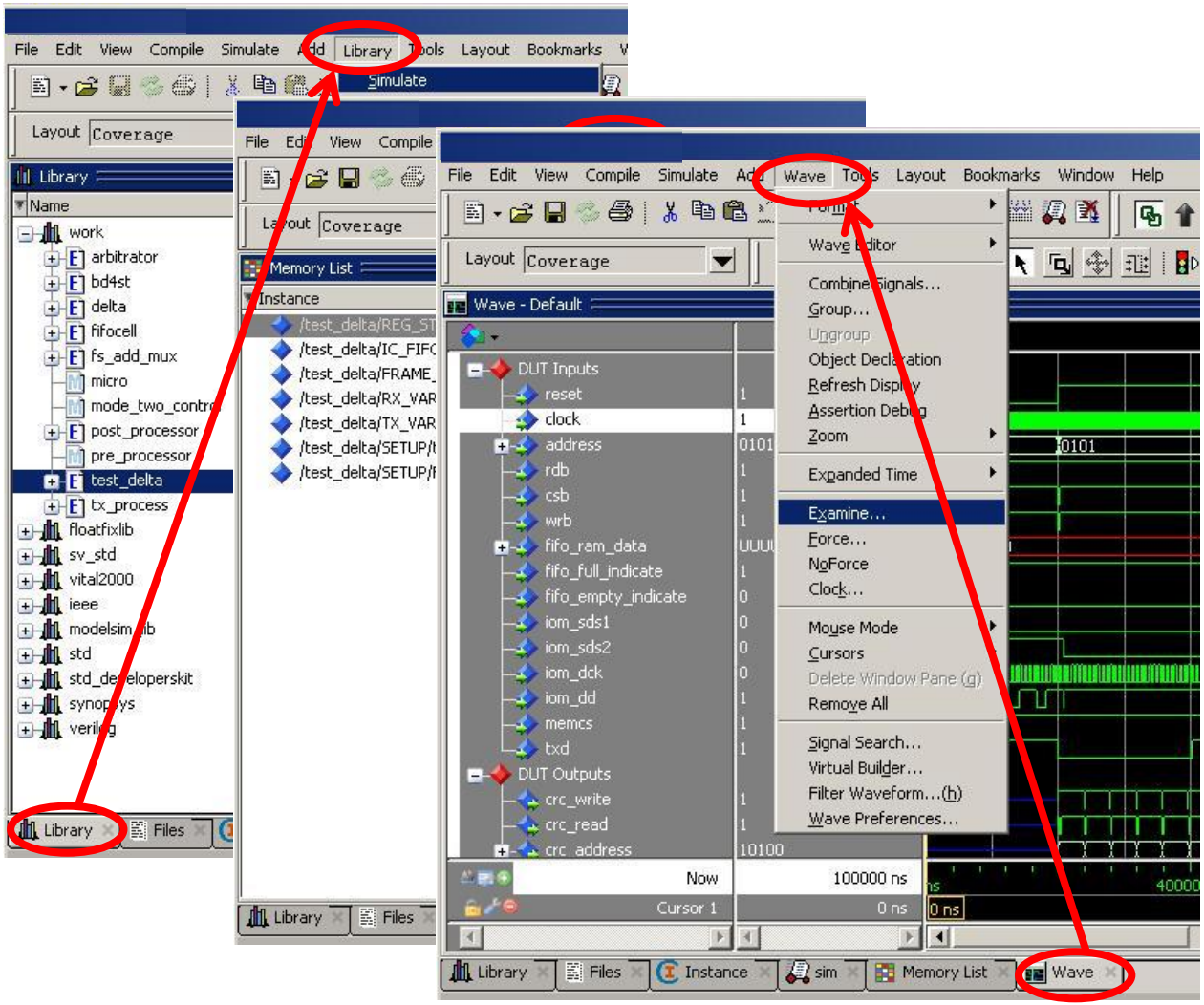


Hint: If you don't want the tool saving all changes to window layouts when you exit, uncheck these boxes

© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Context Sensitive Menus

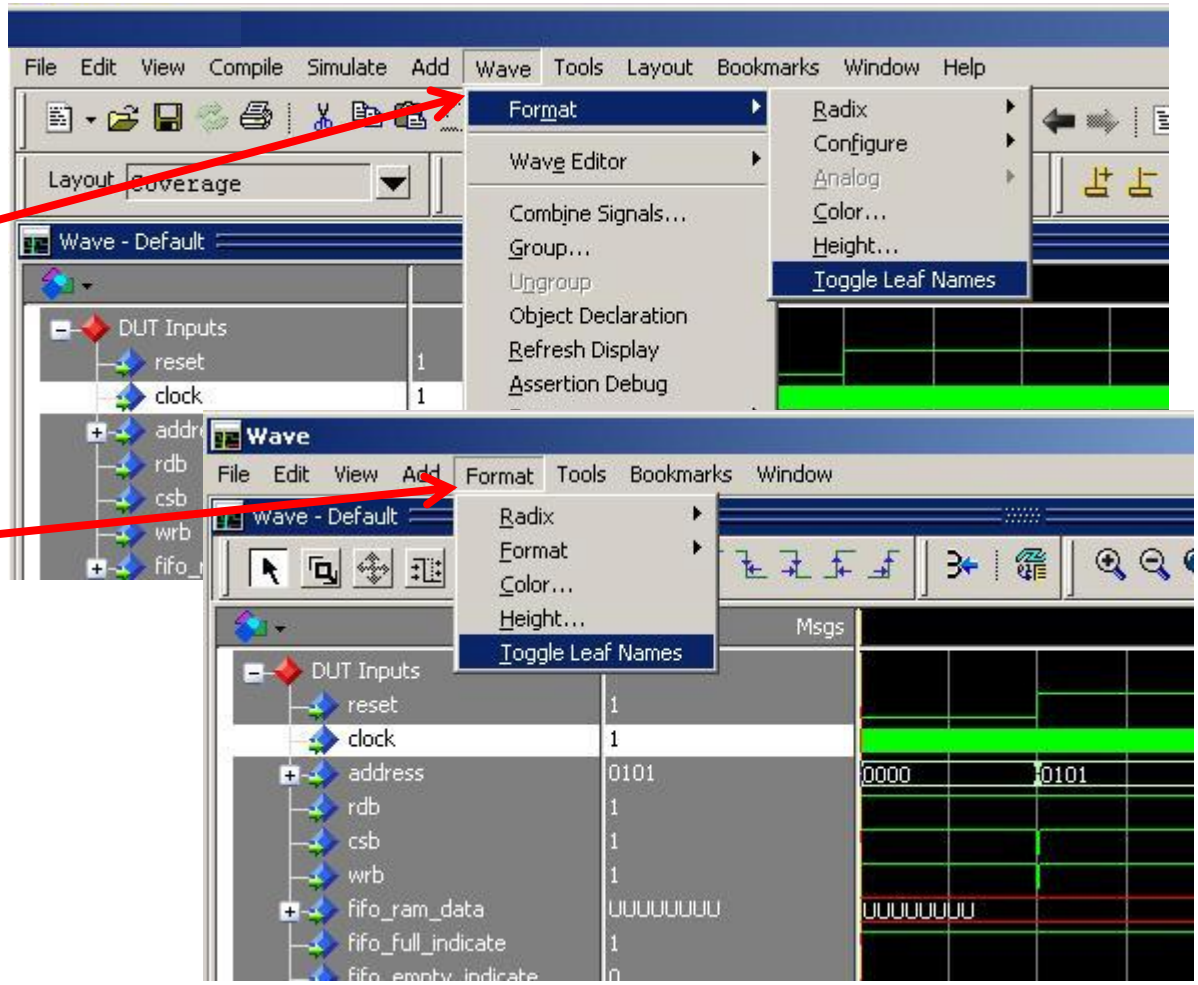
- The main pulldown menu selections are based upon the current "active" window
- Selecting a different window will activate this and add a specific menu pick



© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Docked and Undocked Menus

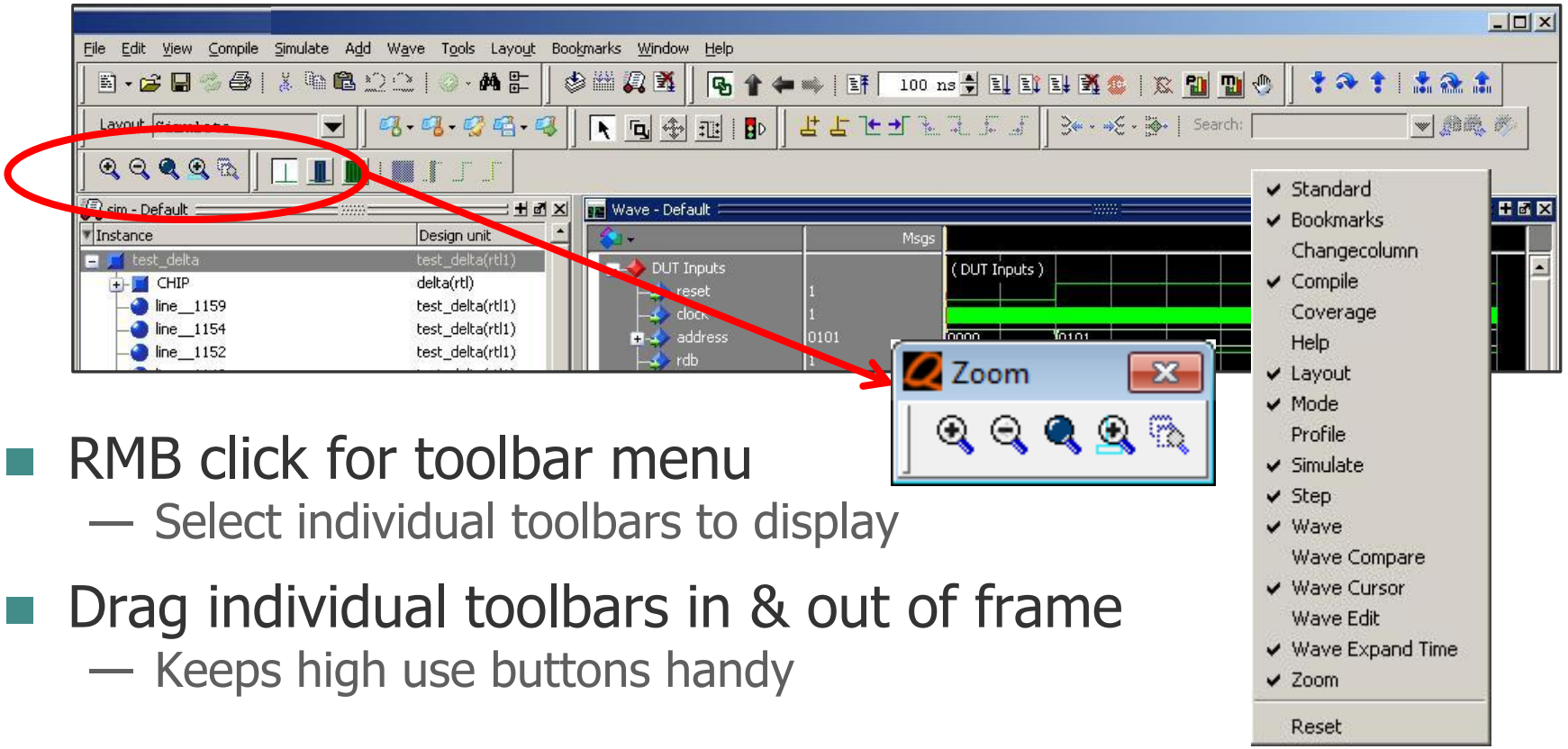
- Menu options are based upon the current "active" docked window
- When undocked the same options are available but under multiple menus



© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Questa's Toolbar

- Toolbar has controls for many common simulation tasks
  - Compile, simulate, run, restart, step, zoom, etc.

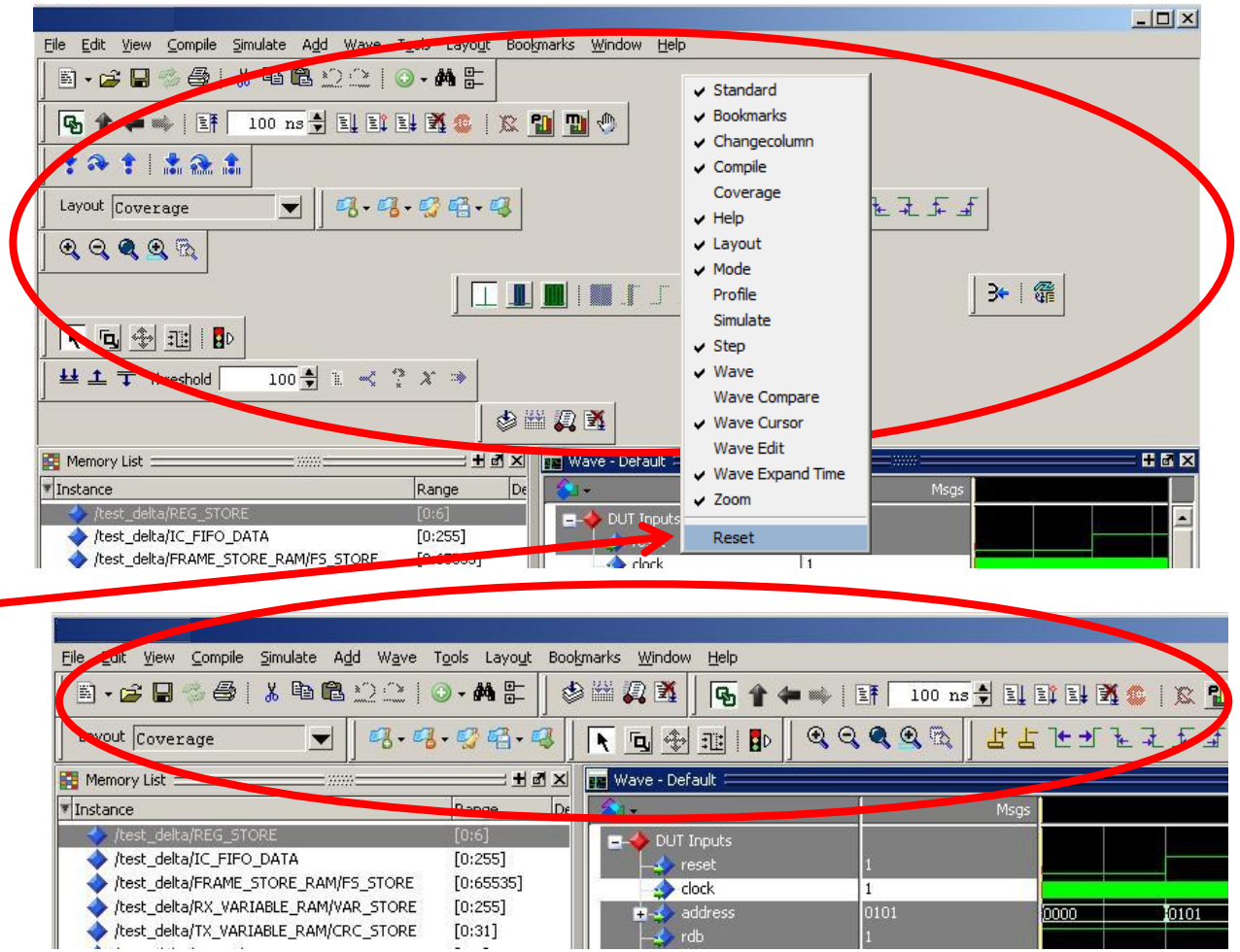


- RMB click for toolbar menu
  - Select individual toolbars to display
- Drag individual toolbars in & out of frame
  - Keeps high use buttons handy

© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Resetting the Toolbar

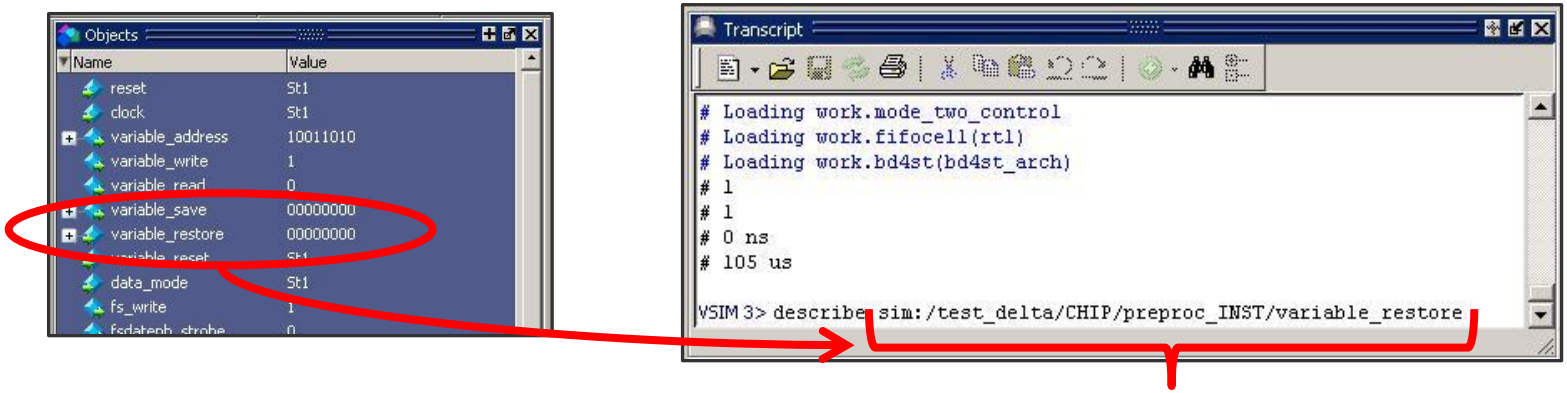
- The menu groups within the toolbar often get "jumbled", especially after the window is repeatedly resized
- RMB in menu area and select "Reset" to tidy this up
- This will re-pack the menu groups



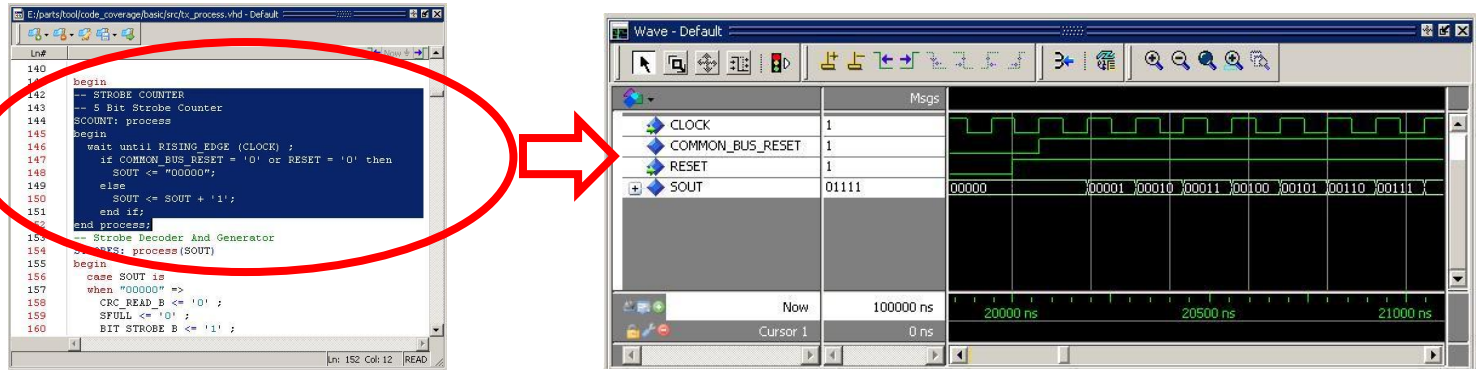
© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Drag & Drop

- Drag & Drop works between every window
- Dragging an any object to the transcript prints full path



- Dragging blocks of code adds all signals

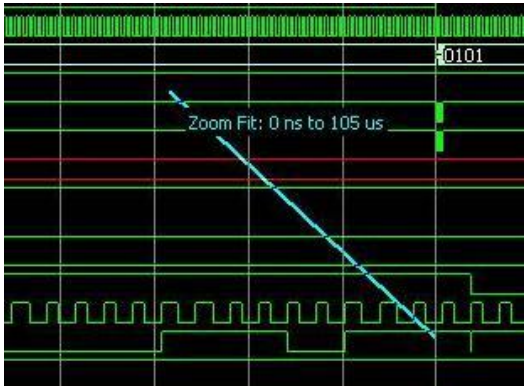


© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

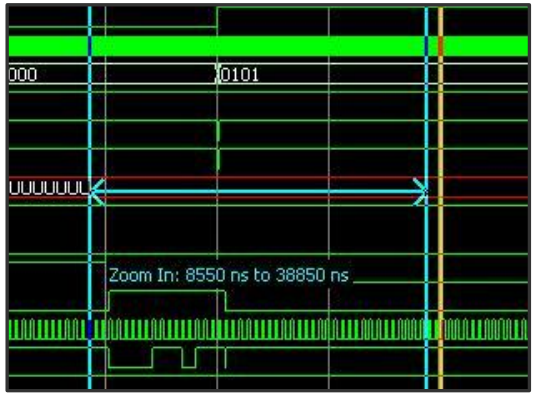
Strokes

- Mouse strokes are a quick way to execute zooming functions in a window
  - Supported in wave & dataflow windows
  - Functions slightly different depending on window
  - Functions labeled for easy identification
- Just hold down MMB and drag in any direction

Hint: If you don't have a 3-button mouse you can press the left & right button simultaneously



Zoom Full:
drag up & left



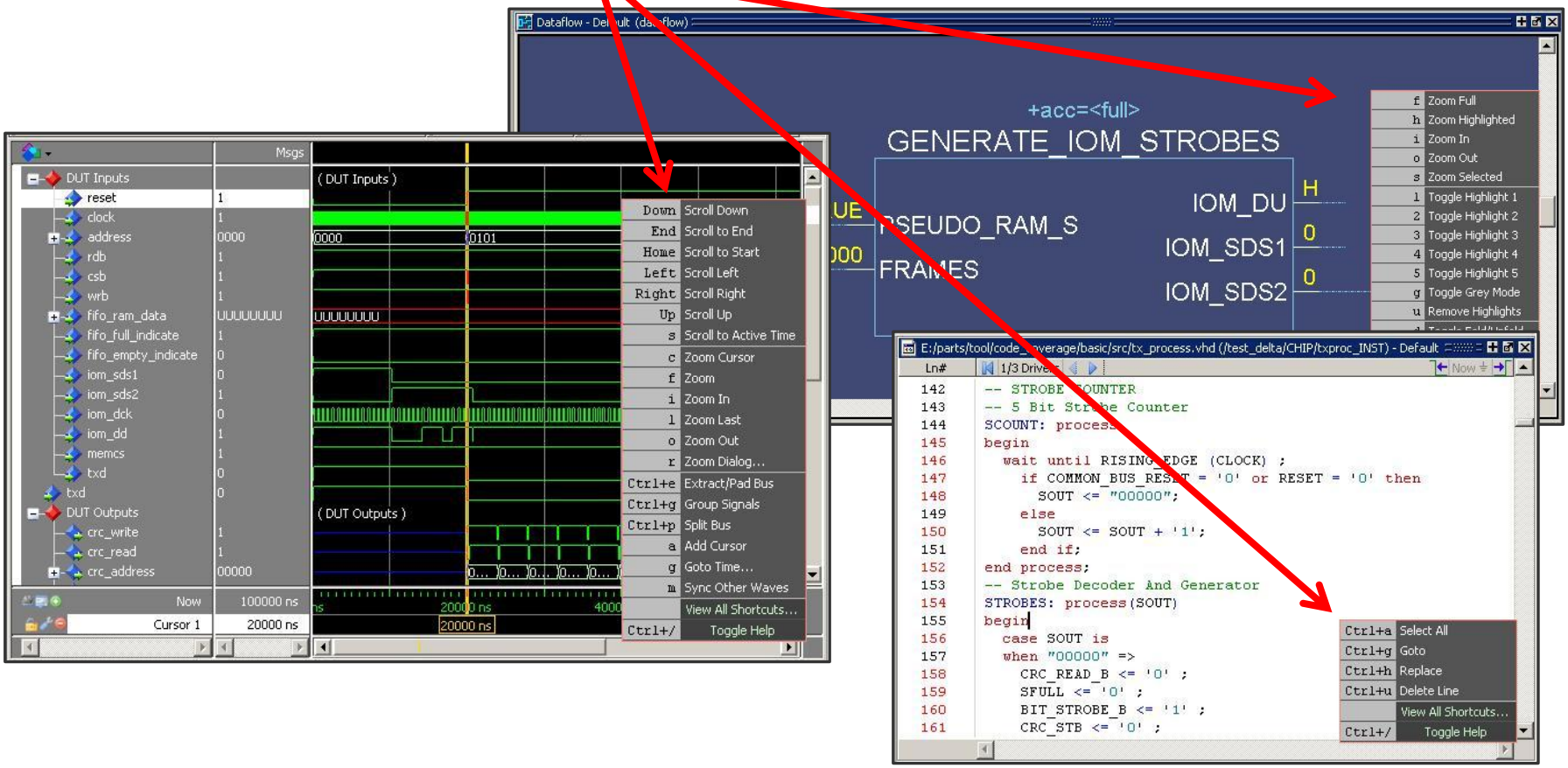
Zoom Area:
drag down & left or
drag down & right

Zoom Out:
drag up & right

© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Keyboard Shortcuts

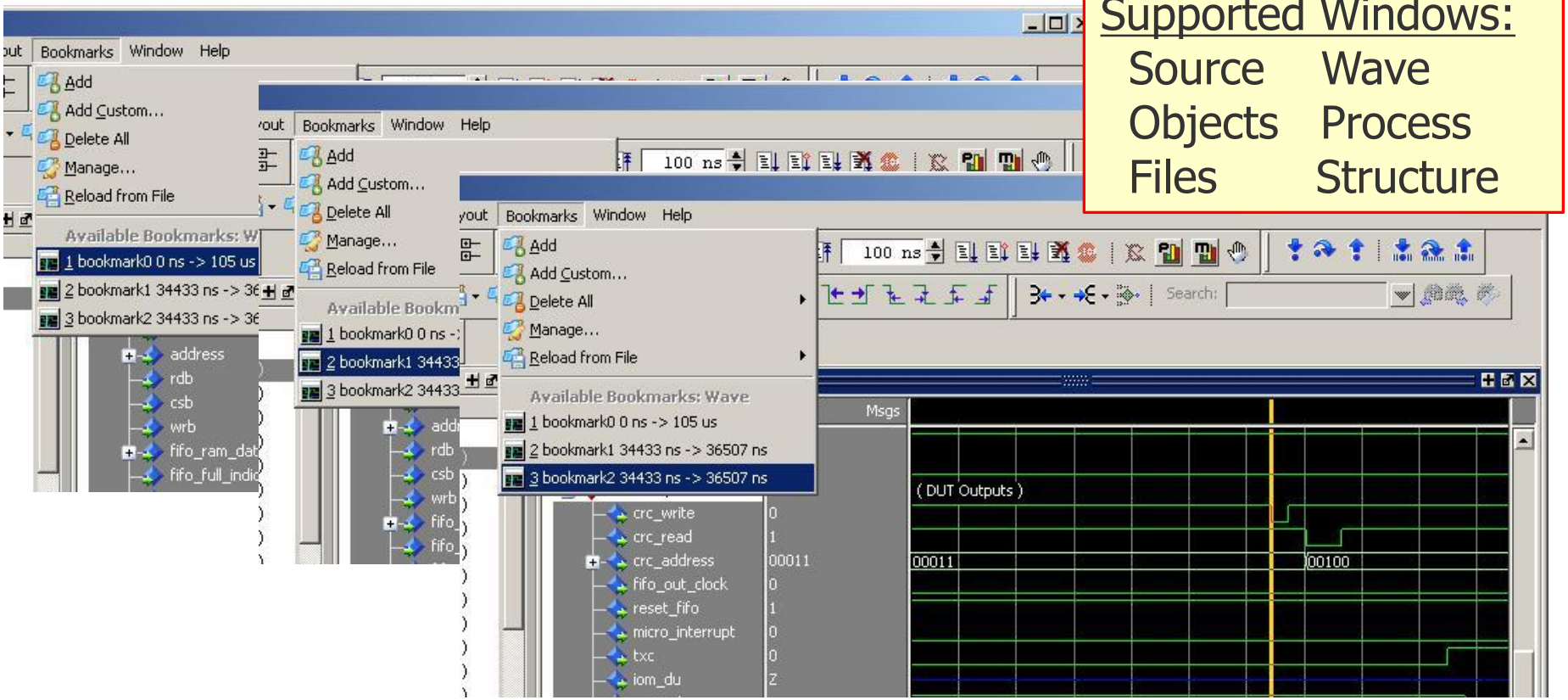
- Each window has a variety of keyboard shortcuts
  - Toggle menu ON/OFF using Ctrl+/



© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Bookmarks

- Bookmarks allow you to save different views in the GUI
  - Quickly move between these different views
  - Bookmarks done on a per window basis



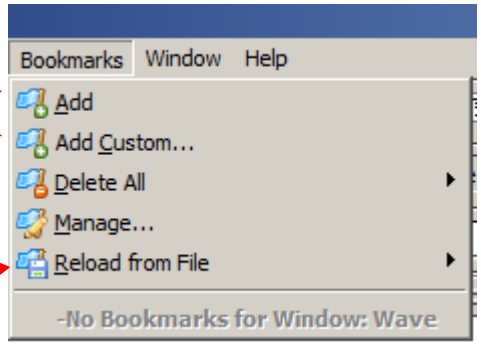
Supported Windows:

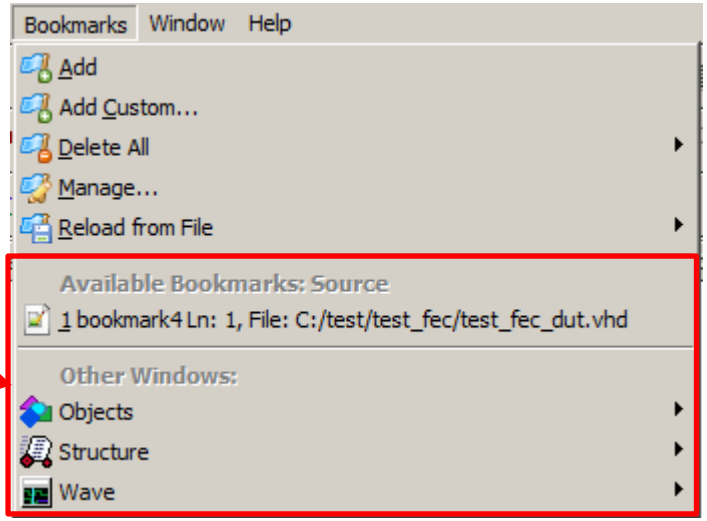
Source	Wave
Objects	Process
Files	Structure

© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Bookmark Menus

- Can be added "as-is" or customized
- Can be reloaded from previous session
- Also access from toolbar
- Shortcuts are automatically added to the pull-down menu

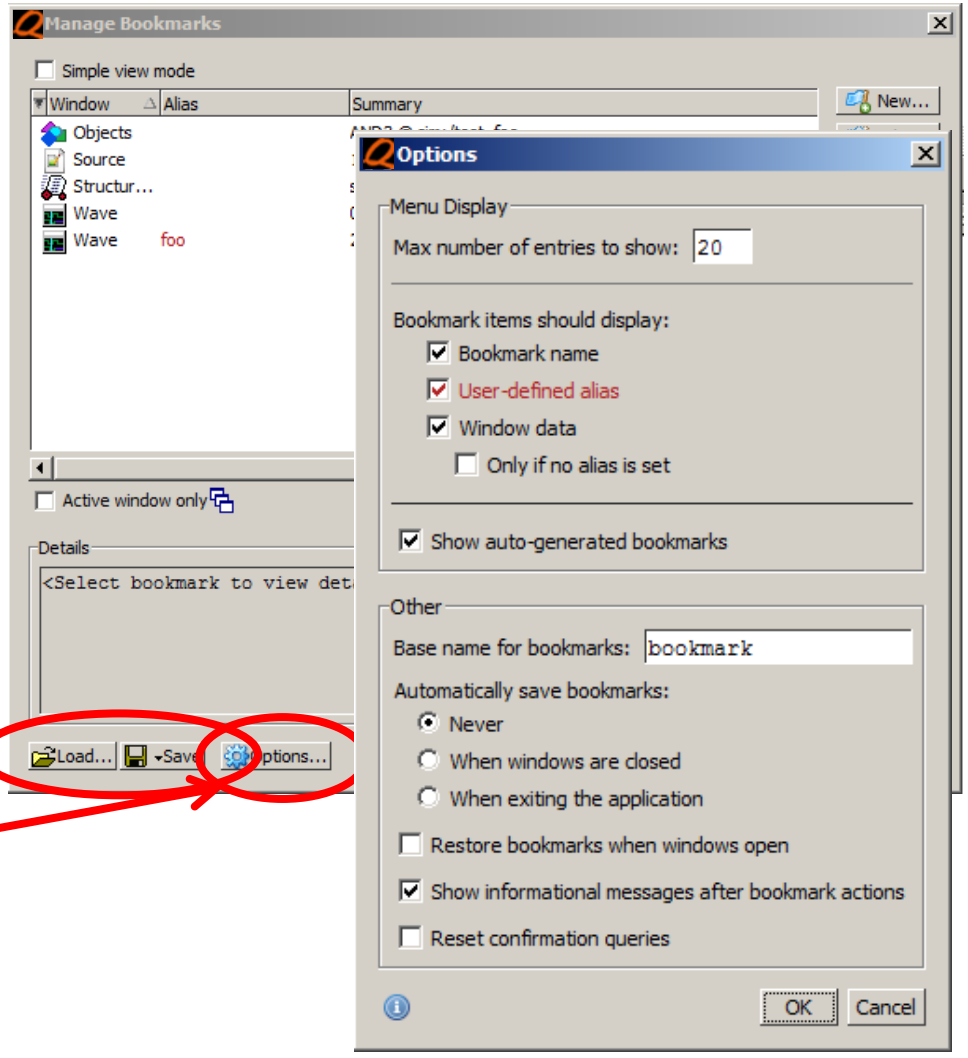




© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Managing Bookmarks

- User has full control over their bookmarks
  - Named (given an alias)
  - Edited
  - Deleted
- Sets of bookmarks can be saved and loaded for re-use
- Auto-save & auto-restore can be selected from Options menu
  - Saved as *bookmark.do*



© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Failure Isolation/Transfer

- Chip level simulation to block level simulation
  - Provide portable debug IP for developer(s)
- Isolate a top or sub-block from an environment
  - Save extended VCD data for top level you want to isolate (c)
- Re-simulate only sub-block with extended VCD (eVCD) file
  - Best debug throughput with smaller block
- Improved throughput for team responsible for bug fix
  - Use bookmarks and wave.do scripts to paint picture for them

```
VSIM 1> vcd dumpports –file c.vcd /top/c/*
VSIM 2> run -all
```

c

```
csh%> vsim c –vcdstim c.vcd
```


© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

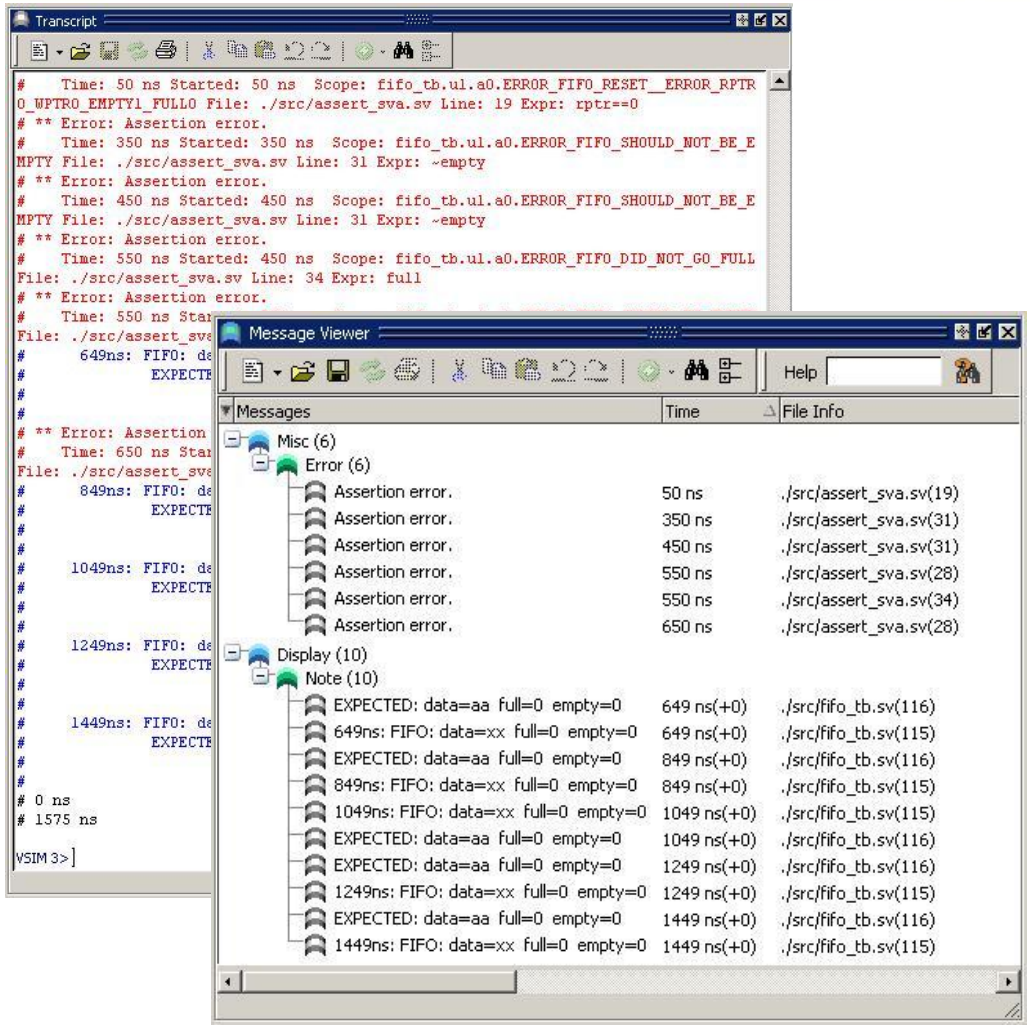
Agenda: Questa Debug Essentials

- General Questa Operations
- Managing Simulation Messages
- Source Level Debugging
- Viewing and Working with Waveforms
- Root Cause Analysis

© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

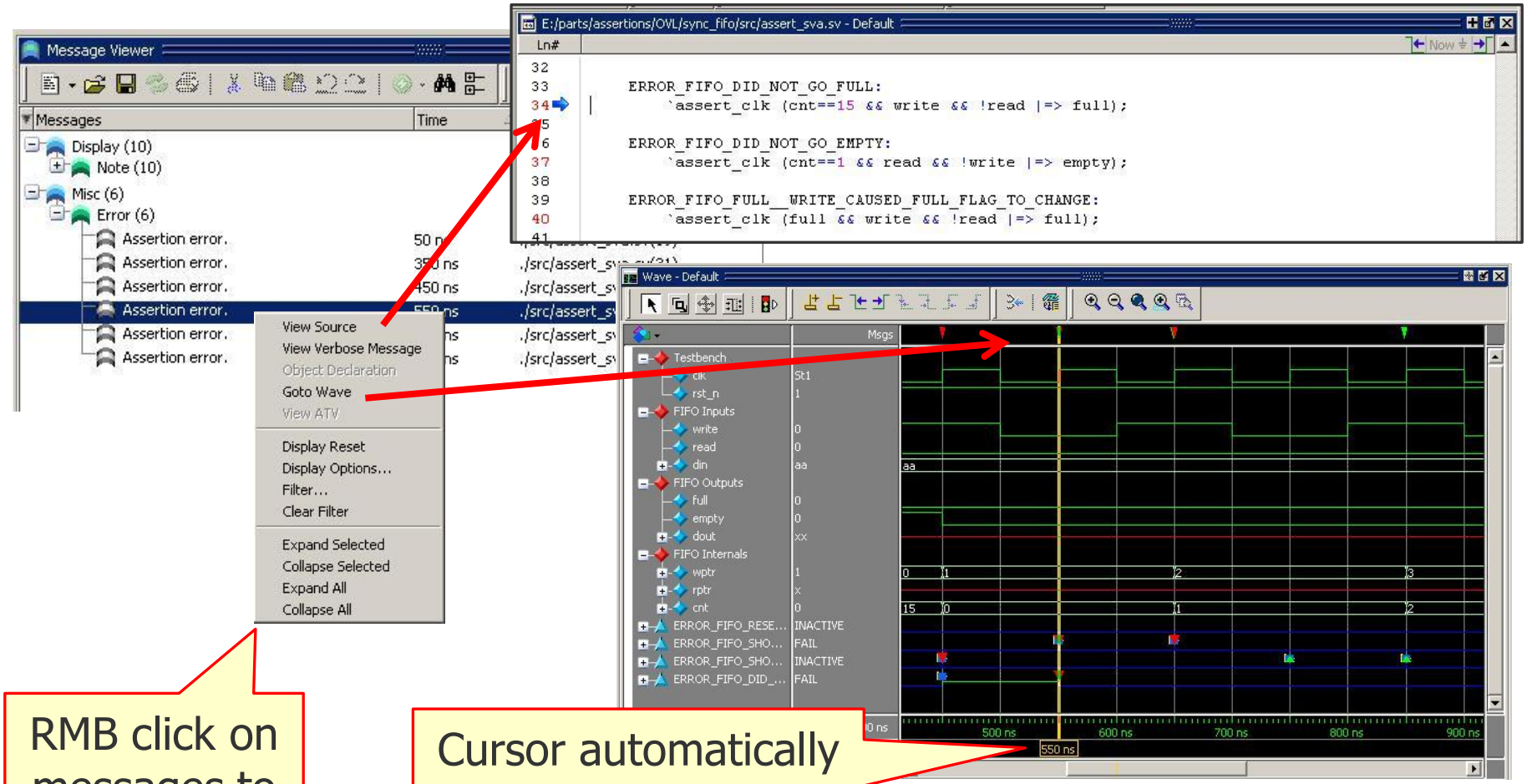
Simulation Messages

- Transcript Window
  - A running history of commands executed
  - Simulation output as it occurs
- Message Viewer
  - Organized by message type and severity
  - Can be filtered and sorted for ease of debugging
  - Linked to both the wave & source windows


Transcript
Time: 50 ns Started: 50 ns Scope: fifo_tb.ul.a0.ERROR_FIFO_RESET__ERROR_RPTR 0_WPTR0_EMPTY1_FULL0 File: ./src/assert_sva.sv Line: 19 Expr: rptr==0
** Error: Assertion error.
Time: 350 ns Started: 350 ns Scope: fifo_tb.ul.a0.ERROR_FIFO_SHOULD_NOT_BE_E MPTY File: ./src/assert_sva.sv Line: 31 Expr: ~empty
** Error: Assertion error.
Time: 450 ns Started: 450 ns Scope: fifo_tb.ul.a0.ERROR_FIFO_SHOULD_NOT_BE_E MPTY File: ./src/assert_sva.sv Line: 31 Expr: ~empty
** Error: Assertion error.
Time: 550 ns Started: 450 ns Scope: fifo_tb.ul.a0.ERROR_FIFO_DID_NOT_GO_FULL File: ./src/assert_sva.sv Line: 34 Expr: full
** Error: Assertion error.
Time: 550 ns Star
File: ./src/assert_sva
649ns: FIFO: de
EXPECTE
Time: 650 ns Star
File: ./src/assert_sva
849ns: FIFO: de
EXPECTE
1049ns: FIFO: de
EXPECTE
1249ns: FIFO: de
EXPECTE
1449ns: FIFO: de
EXPECTE
0 ns
1575 ns
VSIM 3>
Message Viewer
Help
Messages
Time
File Info
Misc (6)
Error (6)
Assertion error. 50 ns ./src/assert_sva.sv(19)
Assertion error. 350 ns ./src/assert_sva.sv(31)
Assertion error. 450 ns ./src/assert_sva.sv(31)
Assertion error. 550 ns ./src/assert_sva.sv(28)
Assertion error. 550 ns ./src/assert_sva.sv(34)
Assertion error. 650 ns ./src/assert_sva.sv(28)
Display (10)
Note (10)
EXPECTED: data=aa full=0 empty=0 649 ns(+0) ./src/fifo_tb.sv(116)
649ns: FIFO: data=xx full=0 empty=0 649 ns(+0) ./src/fifo_tb.sv(115)
EXPECTED: data=aa full=0 empty=0 849 ns(+0) ./src/fifo_tb.sv(116)
849ns: FIFO: data=xx full=0 empty=0 849 ns(+0) ./src/fifo_tb.sv(115)
1049ns: FIFO: data=xx full=0 empty=0 1049 ns(+0) ./src/fifo_tb.sv(115)
EXPECTED: data=aa full=0 empty=0 1049 ns(+0) ./src/fifo_tb.sv(116)
EXPECTED: data=aa full=0 empty=0 1249 ns(+0) ./src/fifo_tb.sv(116)
1249ns: FIFO: data=xx full=0 empty=0 1249 ns(+0) ./src/fifo_tb.sv(115)
EXPECTED: data=aa full=0 empty=0 1449 ns(+0) ./src/fifo_tb.sv(116)
1449ns: FIFO: data=xx full=0 empty=0 1449 ns(+0) ./src/fifo_tb.sv(115)

© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Message Viewer to Source or Wave Linking



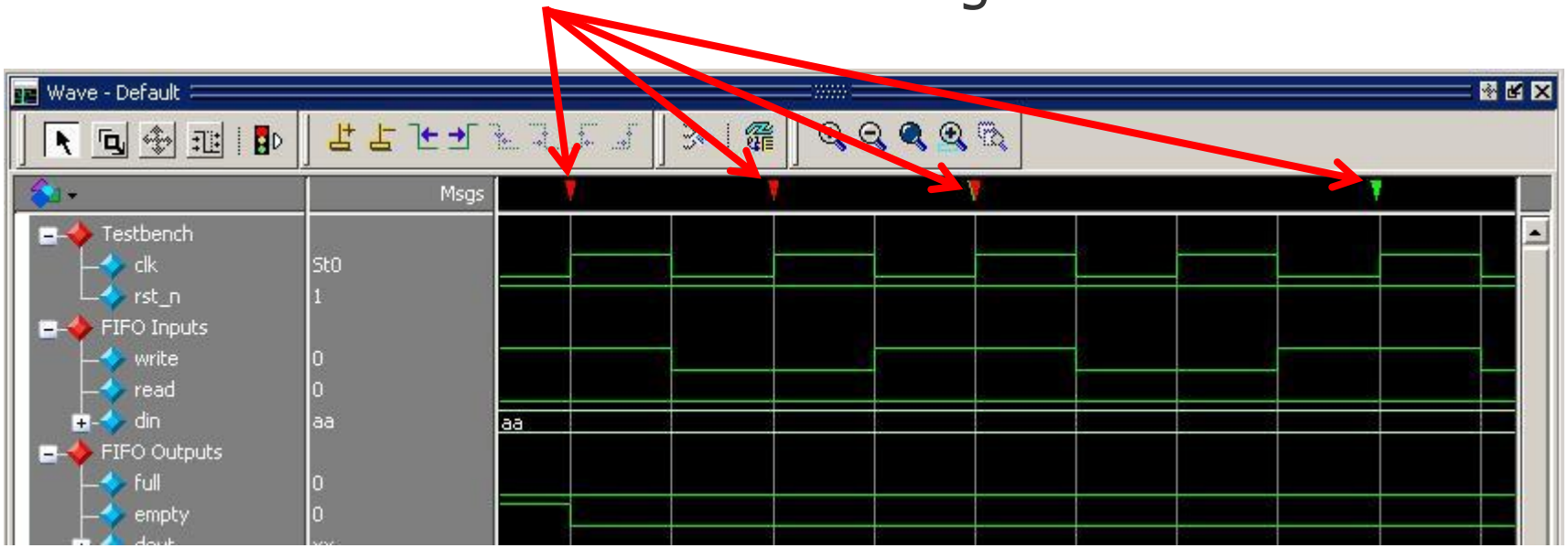
RMB click on messages to see options

Cursor automatically inserted at time message occurred

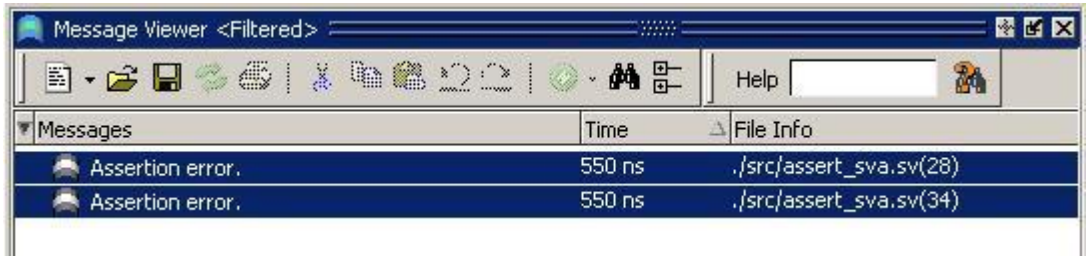
© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Wave Window to Message Viewer Linking

- Wave window marked when messages occurs



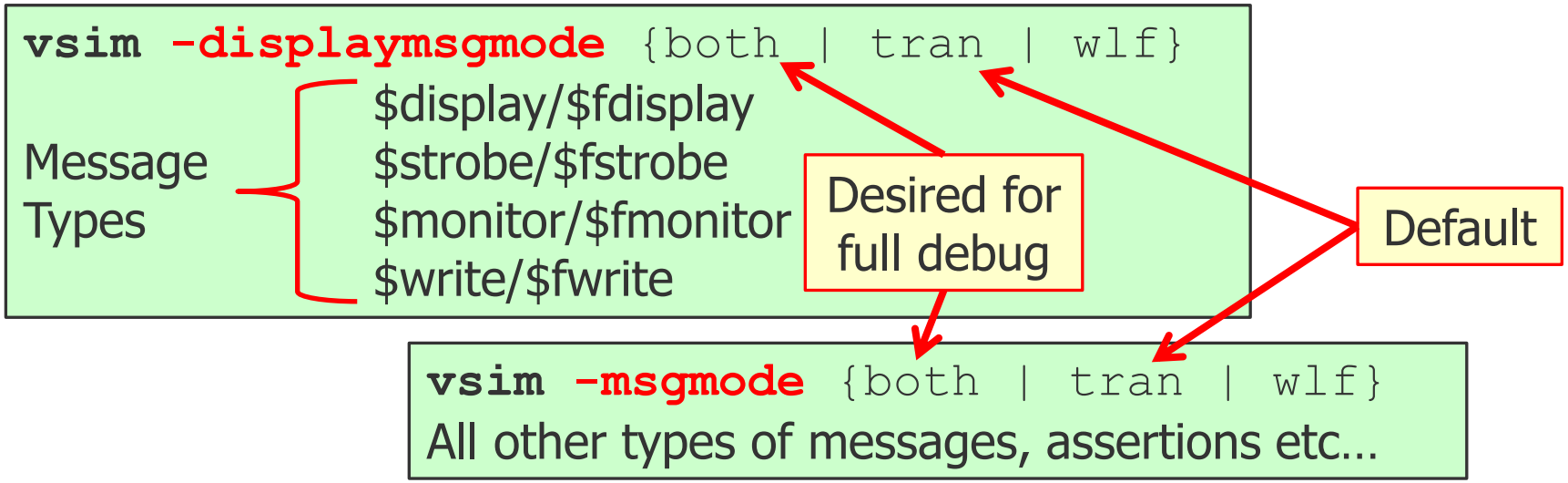
- Double click the message marker to automatically filter the message viewer



© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

How to Enable the Message Viewer

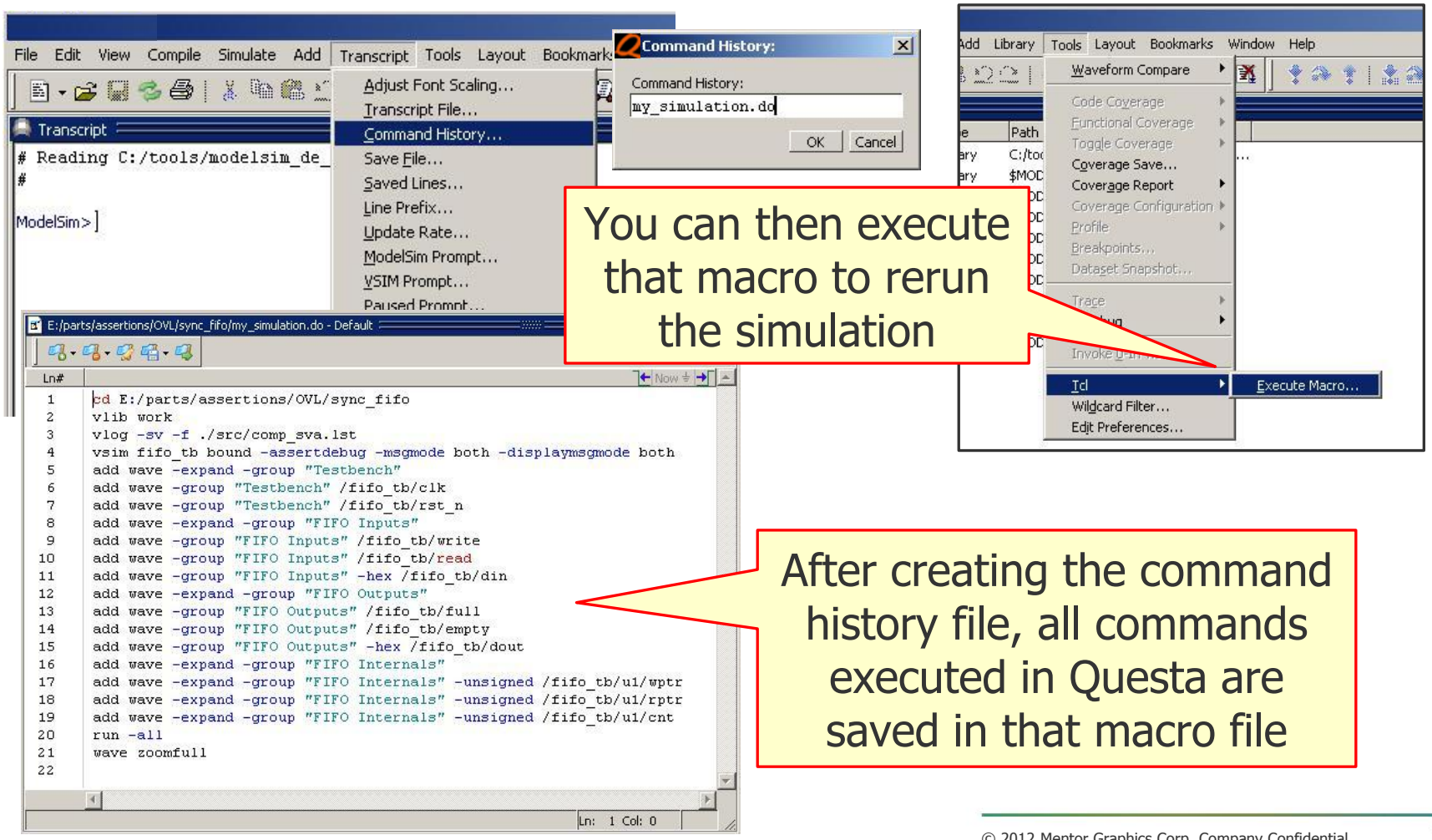
- By default messages are only printed in the transcript
- Message viewer enabled by modelsim.ini variable or command line switch
  - Provides message viewing in an organized and filterable form
  - Saves messages for post-processing debug
  - Provides links in the wave window
  - Two types of messages



© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Creating a Macro from Executed Commands

- Enabling Command history builds a simulation macro


File Edit View Compile Simulate Add Transcript Tools Layout Bookmarks
Adjust Font Scaling...
Transcript File...
Command History...
Save File...
Saved Lines...
Line Prefix...
Update Rate...
ModelSim Prompt...
VSIM Prompt...
Paused Prompt...
Transcript
Reading C:/tools/modelsim_de_
#
ModelSim>
Command History:
Command History:
my_simulation.do
OK
Cancel
Add Library Tools Layout Bookmarks Window Help
Waveform Compare
Code Coverage
Functional Coverage
Toggle Coverage
Coverage Save...
Coverage Report
Coverage Configuration
Profile
Breakpoints...
Dataset Snapshot...
Trace
Tcl
Wildcard Filter...
Edit Preferences...
Execute Macro...
E:/parts/assertions/OVL/sync_fifo/my_simulation.do - Default
Ln#
1 cd E:/parts/assertions/OVL/sync_fifo
2 vlib work
3 vlog -sv -f ./src/comp_sva.lst
4 vsim fifo_tb bound -assertdebug -msgmode both -displaymsgmode both
5 add wave -expand -group "Testbench"
6 add wave -group "Testbench" /fifo_tb/clk
7 add wave -group "Testbench" /fifo_tb/rst_n
8 add wave -expand -group "FIFO Inputs"
9 add wave -group "FIFO Inputs" /fifo_tb/write
10 add wave -group "FIFO Inputs" /fifo_tb/read
11 add wave -group "FIFO Inputs" -hex /fifo_tb/din
12 add wave -expand -group "FIFO Outputs"
13 add wave -group "FIFO Outputs" /fifo_tb/full
14 add wave -group "FIFO Outputs" /fifo_tb/empty
15 add wave -group "FIFO Outputs" -hex /fifo_tb/dout
16 add wave -expand -group "FIFO Internals"
17 add wave -expand -group "FIFO Internals" -unsigned /fifo_tb/u1/wptr
18 add wave -expand -group "FIFO Internals" -unsigned /fifo_tb/u1/rptr
19 add wave -expand -group "FIFO Internals" -unsigned /fifo_tb/u1/cnt
20 run -all
21 wave zoomfull
22
Ln: 1 Col: 0
You can then execute that macro to rerun the simulation
After creating the command history file, all commands executed in Questa are saved in that macro file

© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Agenda: Questa Debug Essentials

- General Questa Operations
- Managing Simulation Messages
- Source Level Debugging
- Viewing and Working with Waveforms
- Root Cause Analysis

© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

The Source Window

- The source window in Questa serves many purposes

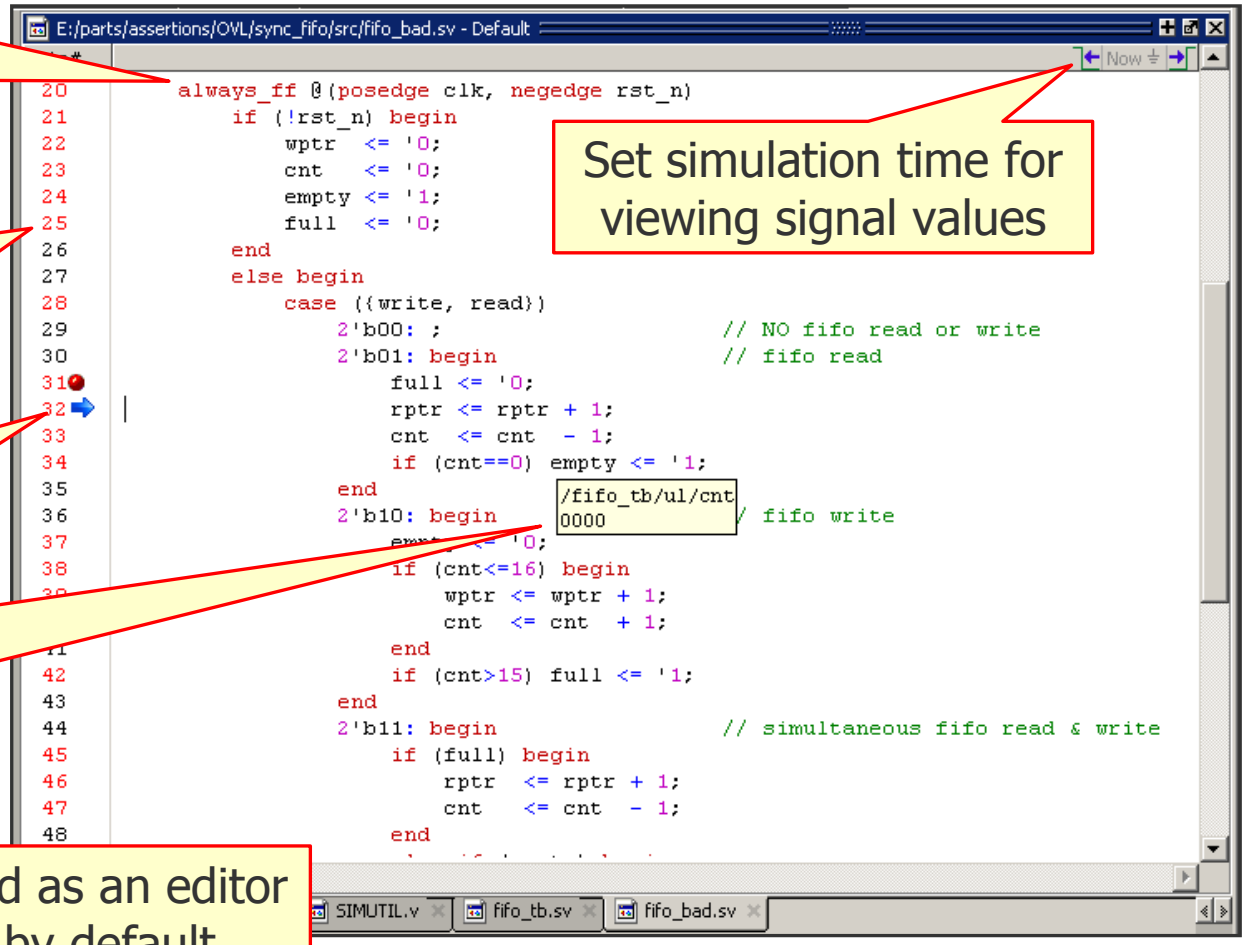
Source code displayed with syntax highlighting

Red line numbers indicate executable lines

Breakpoints and current line indicator

Balloon popups show signal values or use source code annotation

Set simulation time for viewing signal values



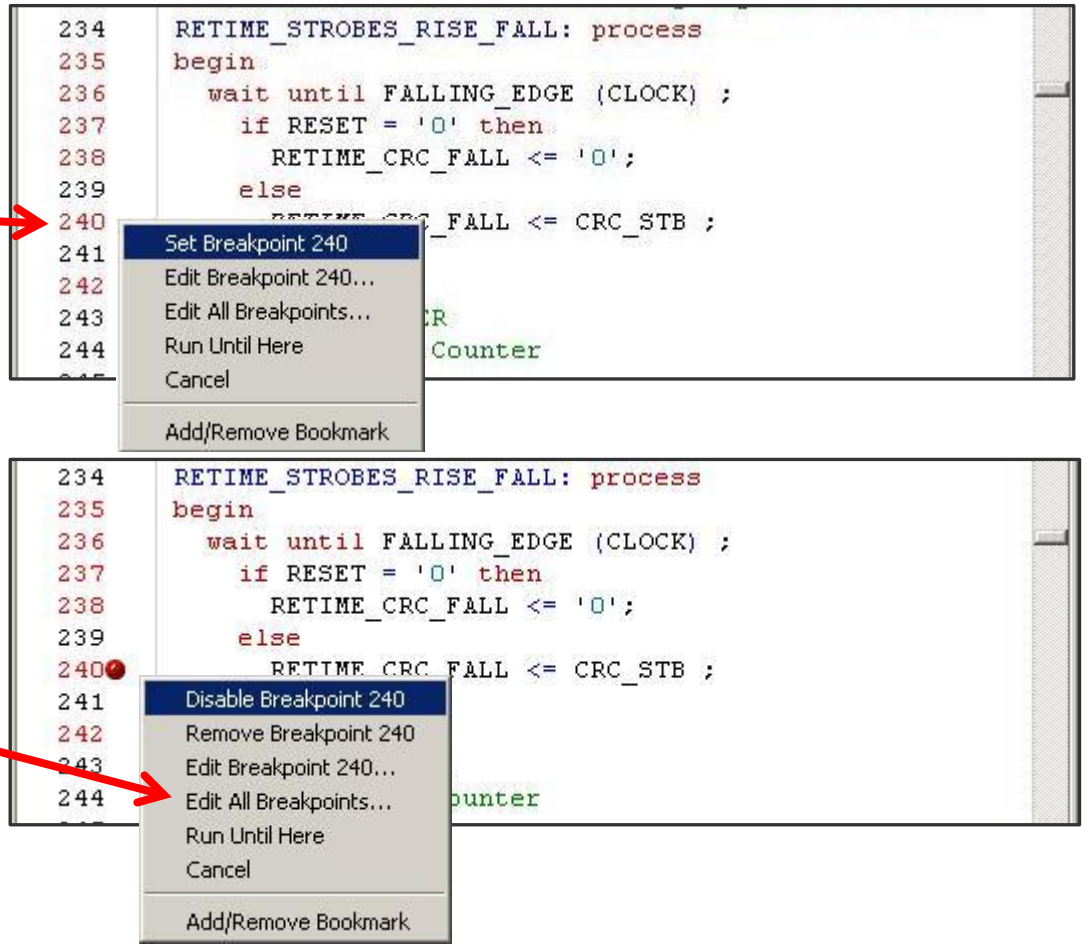

```
20      always_ff @(posedge clk, negedge rst_n)
21          if (!rst_n) begin
22              wptr  <= '0;
23              cnt   <= '0;
24              empty <= '1;
25              full  <= '0;
26          end
27          else begin
28              case ({write, read})
29                  2'b00: ;                        // NO fifo read or write
30                  2'b01: begin                    // fifo read
31                      full <= '0;
32                      rptr <= rptr + 1;
33                      cnt  <= cnt  - 1;
34                      if (cnt==0) empty <= '1;
35                  end
36                  2'b10: begin                    // fifo write
37                      empty <= '0;
38                      if (cnt<=16) begin
                            wptr <= wptr + 1;
                            cnt  <= cnt  + 1;
                        end
42                      if (cnt>15) full <= '1;
43                  end
44                  2'b11: begin                    // simultaneous fifo read & write
45                      if (full) begin
46                          rptr  <= rptr + 1;
47                          cnt   <= cnt  - 1;
48                      end
```

Can also be used as an editor "Read Only" by default

© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Setting Breakpoints

- Breakpoints can be set on any executable line
  - shown with **red** line numbers
- BP shown as red ball
  - LMB click to add BP
  - RMB click for menu
- RMB menu to manage breakpoints

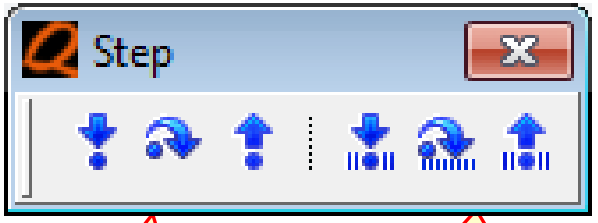


© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Single Stepping Source Code

- Once the tool is stopped at a breakpoint you can walk through the source a line at a time
- Several different types of stepping
  - **Step Into** process
  - **Step Over** process
  - **Step Out** of process

```
234   RETIME_STROBES_RISE_FALL: process
235   begin
236     wait until FALLING_EDGE (CLOCK) ;
237       if RESET = '0' then
238         RETIME_CRC_FALL <= '0';
239       else
240         RETIME_CRC_FALL <= CRC_STB ;
241       end if;
242   end process;
243   -- COLUMN COUNTER
244   -- 5 bit Column Counter
245   CCOUNT: process
```

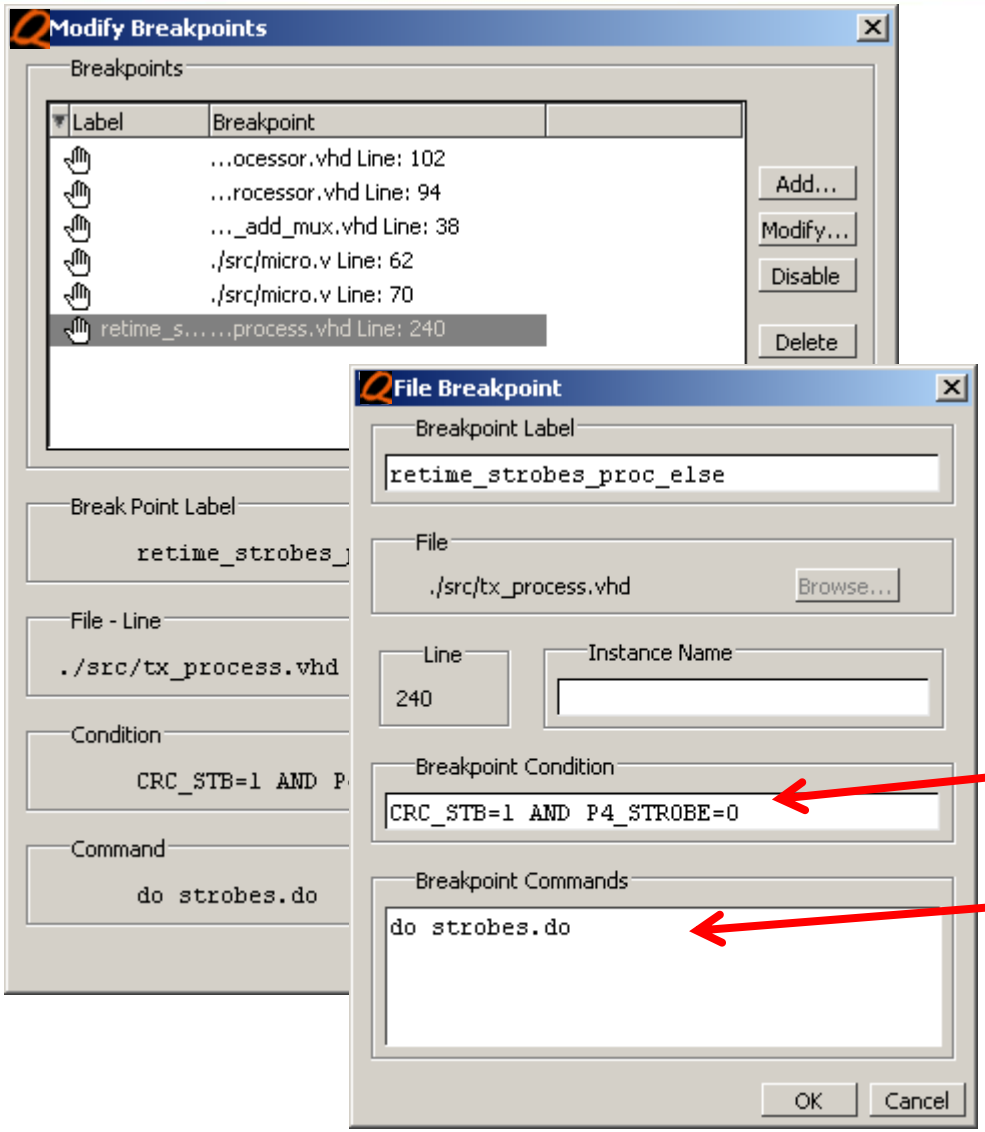


Normal Flow: Follow execution order of simulator regardless of threading

Thread Aware Flow: Stay in the current thread

© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Managing Breakpoints

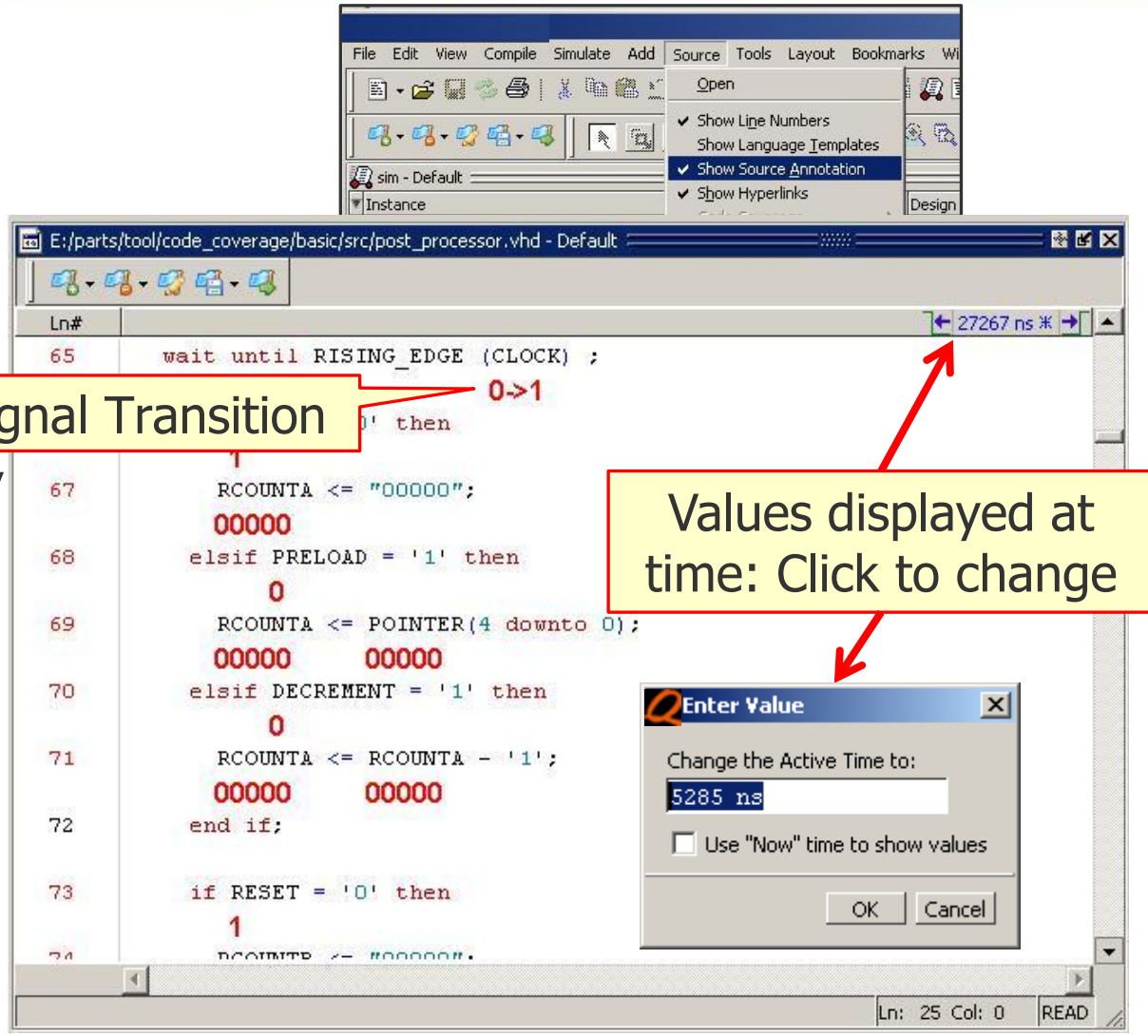


- Full control of each BP in design
  - Add, modify, disable or delete
  - BP can be saved for later reload
- Each BP can have several attributes that controls its behavior
  - Condition when BP happens
  - Commands to execute when BP is hit

© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Source Code Annotation

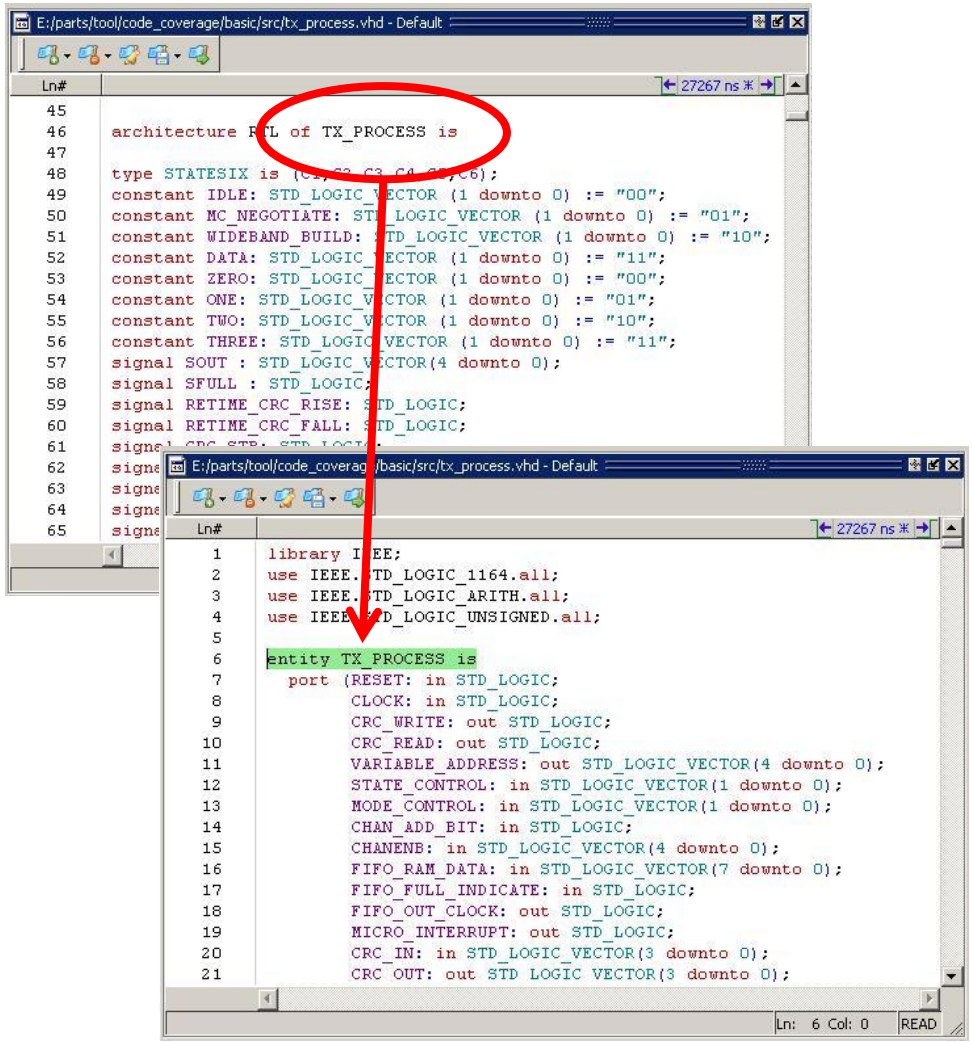
- Maximize amount of information in one window
  - Minimize window switching
- Can be set to display value at any simulation time
- Can be linked to the active cursor in wave window



© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Source Hyperlinking

- Hyperlinking provides easy navigation through source
- Double click when pointer changes to pointing finger ☞
  - Can also set preferences to underline hyperlinks
- Jump from:
  - Usage of a signal, parameter, macro or variable to its declaration
  - VHDL architecture to its entity
  - Entity/module instance to its declaration


E:/parts/tool/code_coverage/basic/src/tx_process.vhd - Default
Ln#
27267 ns
45
46 architecture RTL of TX_PROCESS is
47
48 type STATESIX is (C1, C2, C3, C4, C5, C6);
49 constant IDLE: STD_LOGIC_VECTOR (1 downto 0) := "00";
50 constant MC_NEGOTIATE: STD_LOGIC_VECTOR (1 downto 0) := "01";
51 constant WIDEBAND_BUILD: STD_LOGIC_VECTOR (1 downto 0) := "10";
52 constant DATA: STD_LOGIC_VECTOR (1 downto 0) := "11";
53 constant ZERO: STD_LOGIC_VECTOR (1 downto 0) := "00";
54 constant ONE: STD_LOGIC_VECTOR (1 downto 0) := "01";
55 constant TWO: STD_LOGIC_VECTOR (1 downto 0) := "10";
56 constant THREE: STD_LOGIC_VECTOR (1 downto 0) := "11";
57 signal SOUT : STD_LOGIC_VECTOR(4 downto 0);
58 signal SFULL : STD_LOGIC;
59 signal RETIME_CRC_RISE: STD_LOGIC;
60 signal RETIME_CRC_FALL: STD_LOGIC;
61
62
63
64
65
E:/parts/tool/code_coverage/basic/src/tx_process.vhd - Default
Ln#
27267 ns
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.all;
3 use IEEE.STD_LOGIC_ARITH.all;
4 use IEEE.STD_LOGIC_UNSIGNED.all;
5
6 entity TX_PROCESS is
7 port (RESET: in STD_LOGIC;
8 CLOCK: in STD_LOGIC;
9 CRC_WRITE: out STD_LOGIC;
10 CRC_READ: out STD_LOGIC;
11 VARIABLE_ADDRESS: out STD_LOGIC_VECTOR(4 downto 0);
12 STATE_CONTROL: in STD_LOGIC_VECTOR(1 downto 0);
13 MODE_CONTROL: in STD_LOGIC_VECTOR(1 downto 0);
14 CHAN_ADD_BIT: in STD_LOGIC;
15 CHANENB: in STD_LOGIC_VECTOR(4 downto 0);
16 FIFO_RAM_DATA: in STD_LOGIC_VECTOR(7 downto 0);
17 FIFO_FULL_INDICATE: in STD_LOGIC;
18 FIFO_OUT_CLOCK: out STD_LOGIC;
19 MICRO_INTERRUPT: out STD_LOGIC;
20 CRC_IN: in STD_LOGIC_VECTOR(3 downto 0);
21 CRC_OUT: out STD_LOGIC_VECTOR(3 downto 0);
Ln: 6 Col: 0
READ

© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

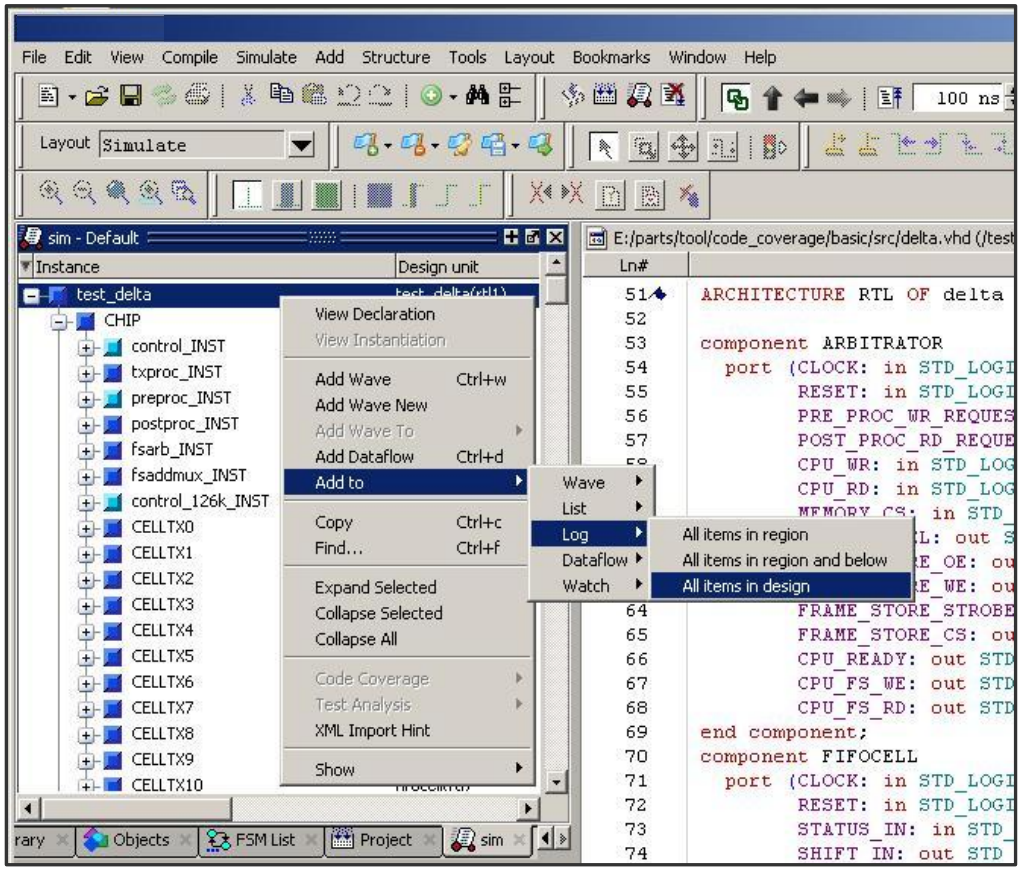
Agenda: Questa Debug Essentials

- General Questa Operations
- Managing Simulation Messages
- Source Level Debugging
- Viewing and Working with Waveforms
- Root Cause Analysis

© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Logging Objects (Signals)

- In order to view a history of an object it must be in the Waveform Log File (WLF) database
- Objects are logged in the WLF any time you add them to the wave/list window or just log them

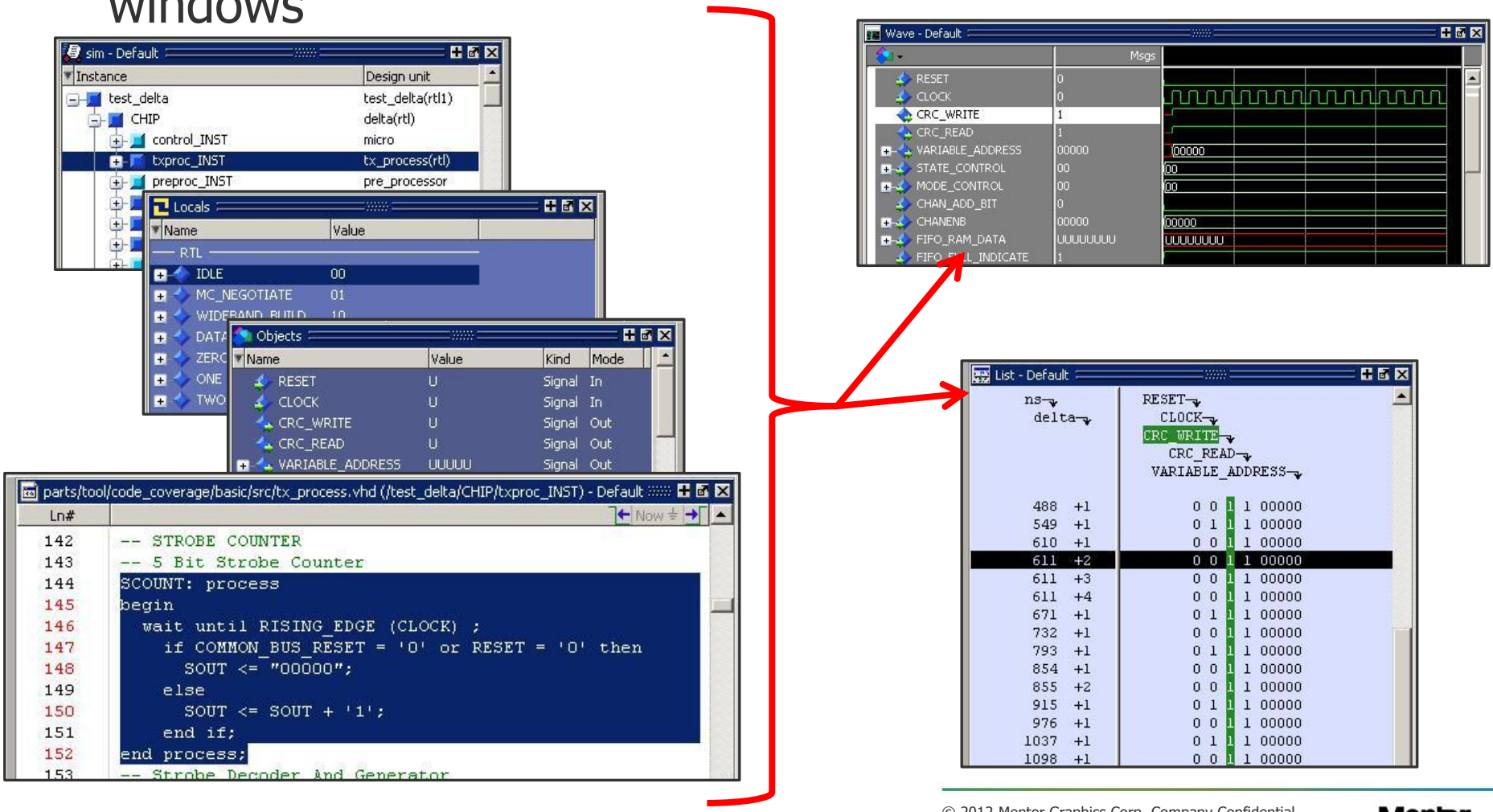


WLF = Wave Log File
Default name = vsim.wlf

© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Adding Signals from Different Windows

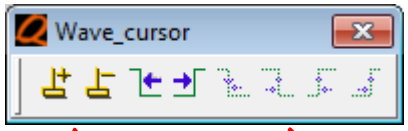
- Adding signals can be done from several different windows



© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Wave Cursors

- Wave cursors have lots ways to control their behavior
- They have many helpful features useful during debug

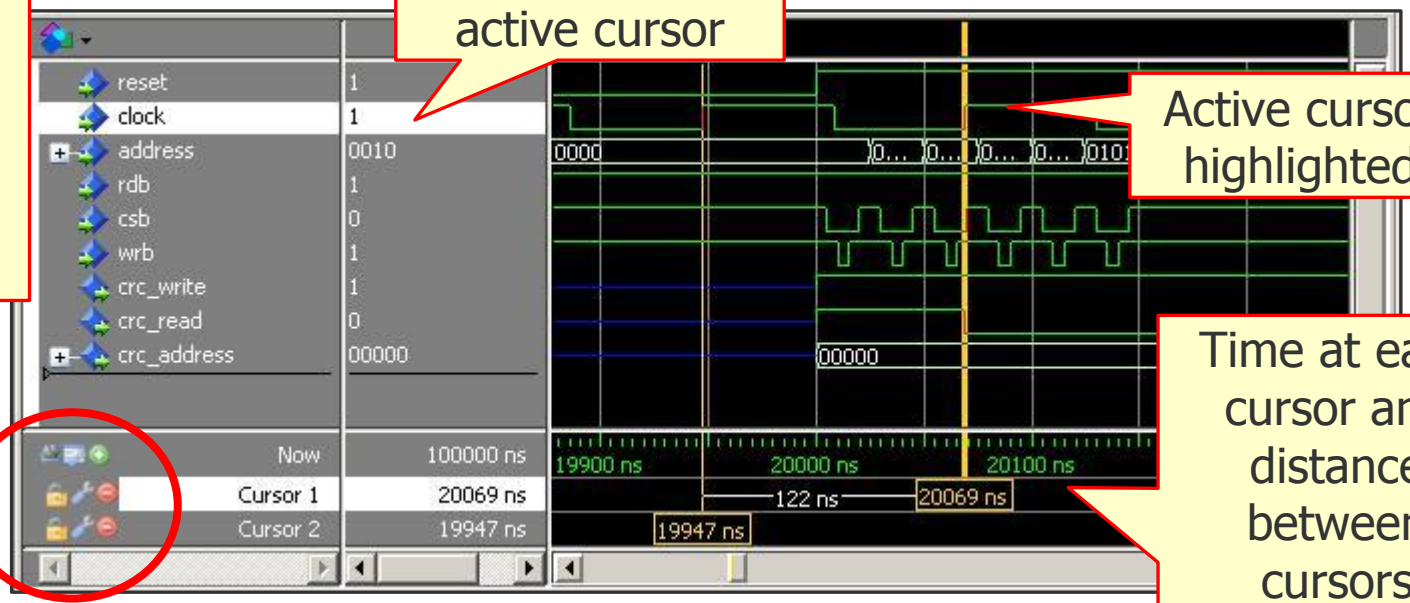


Add/Delete Cursors

Move between edges

Lock/Unlock Properties Add/Remove Gridlines Leaf names

Values linked to active cursor

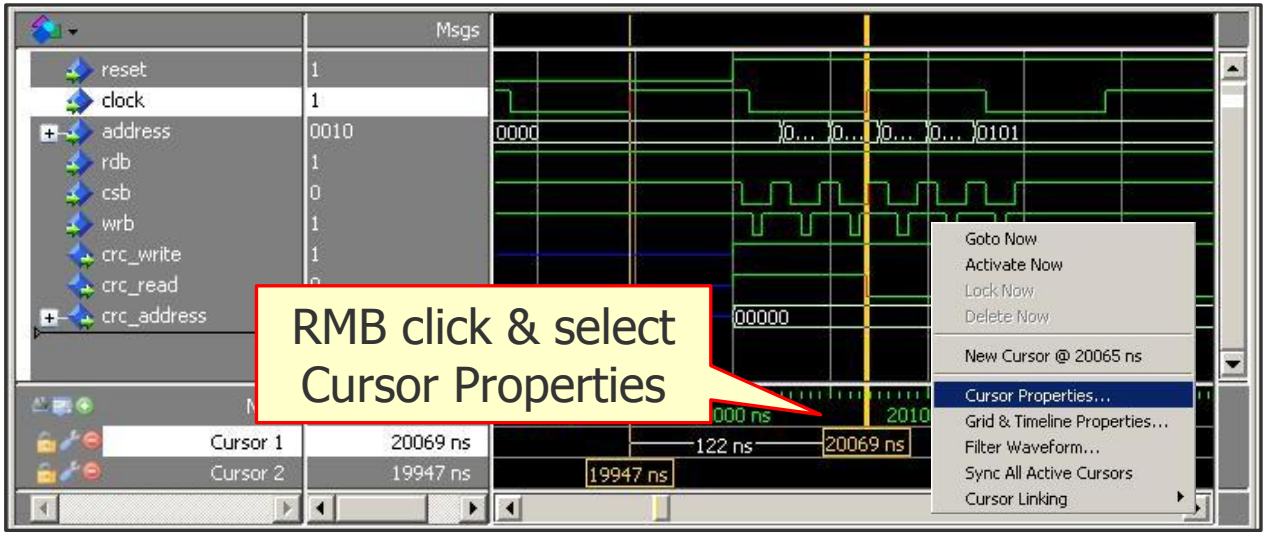


Active cursor highlighted

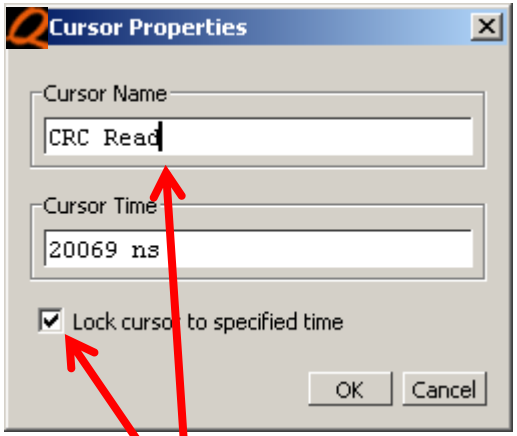
Time at each cursor and distance between cursors

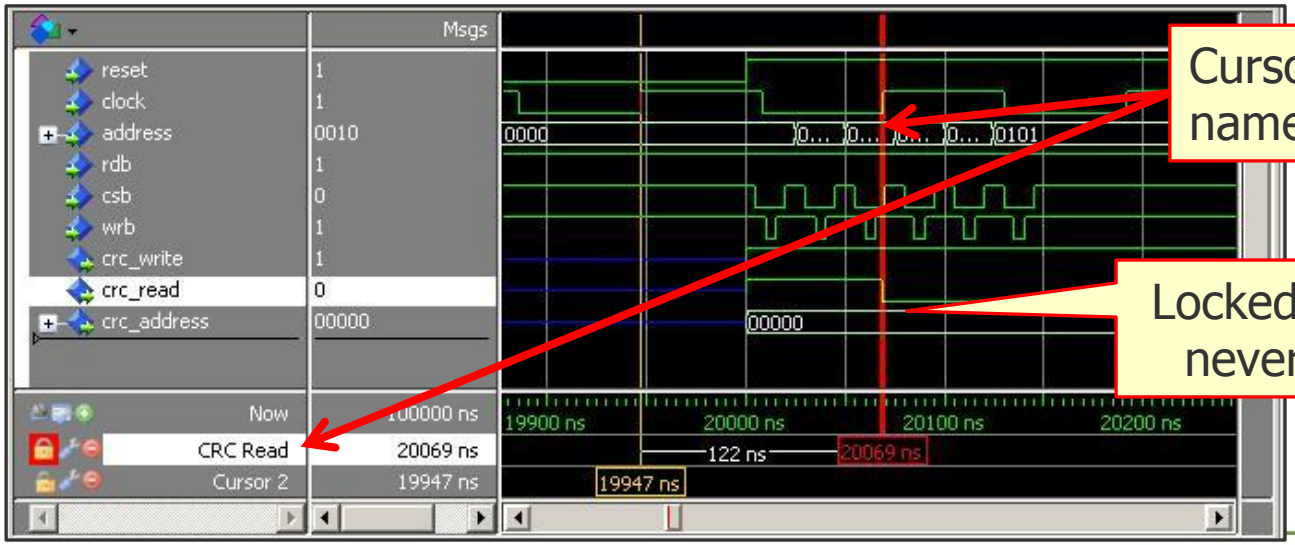
© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Cursor Properties



RMB click & select Cursor Properties



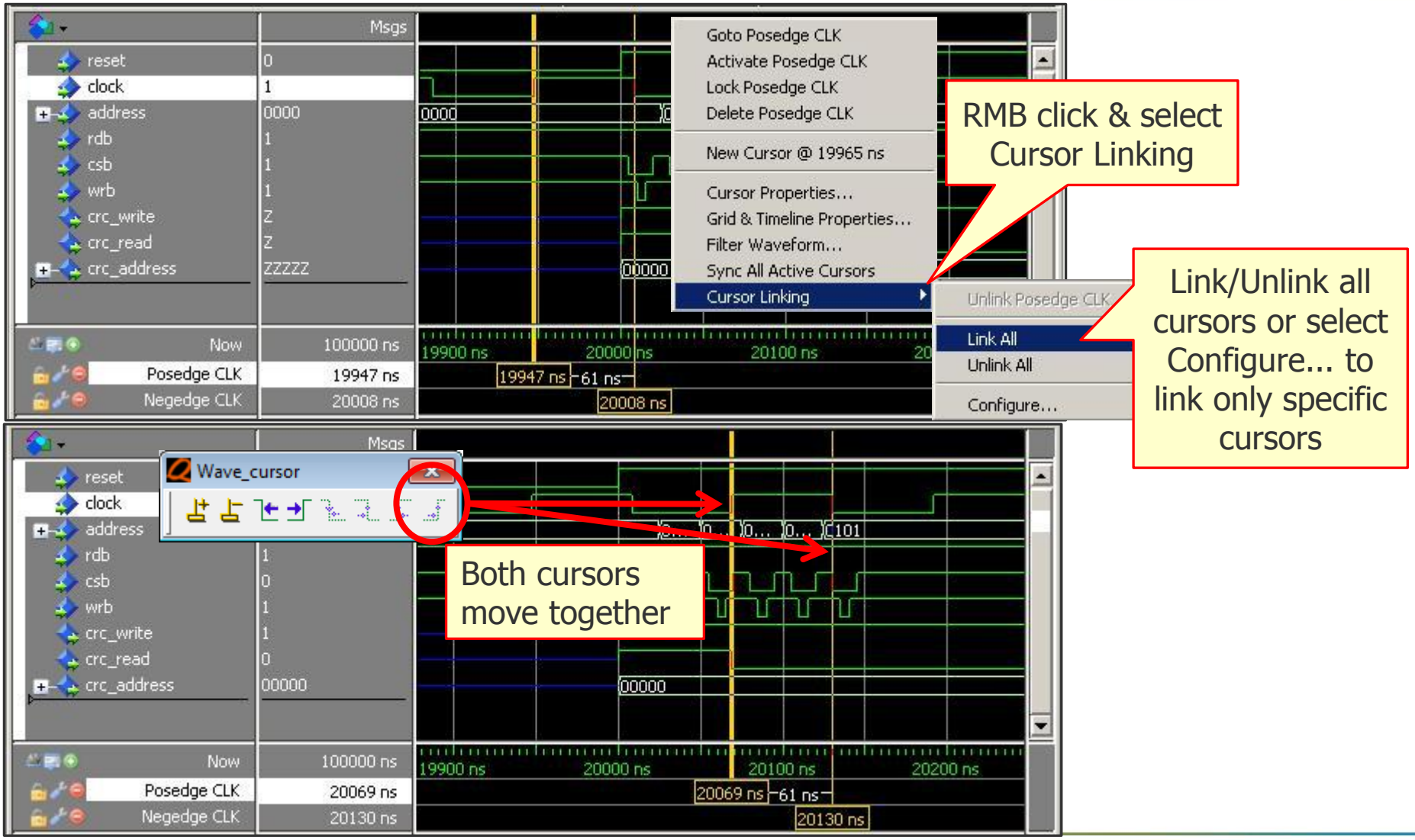


Cursor can be uniquely named and lock in position

Locked cursors never move

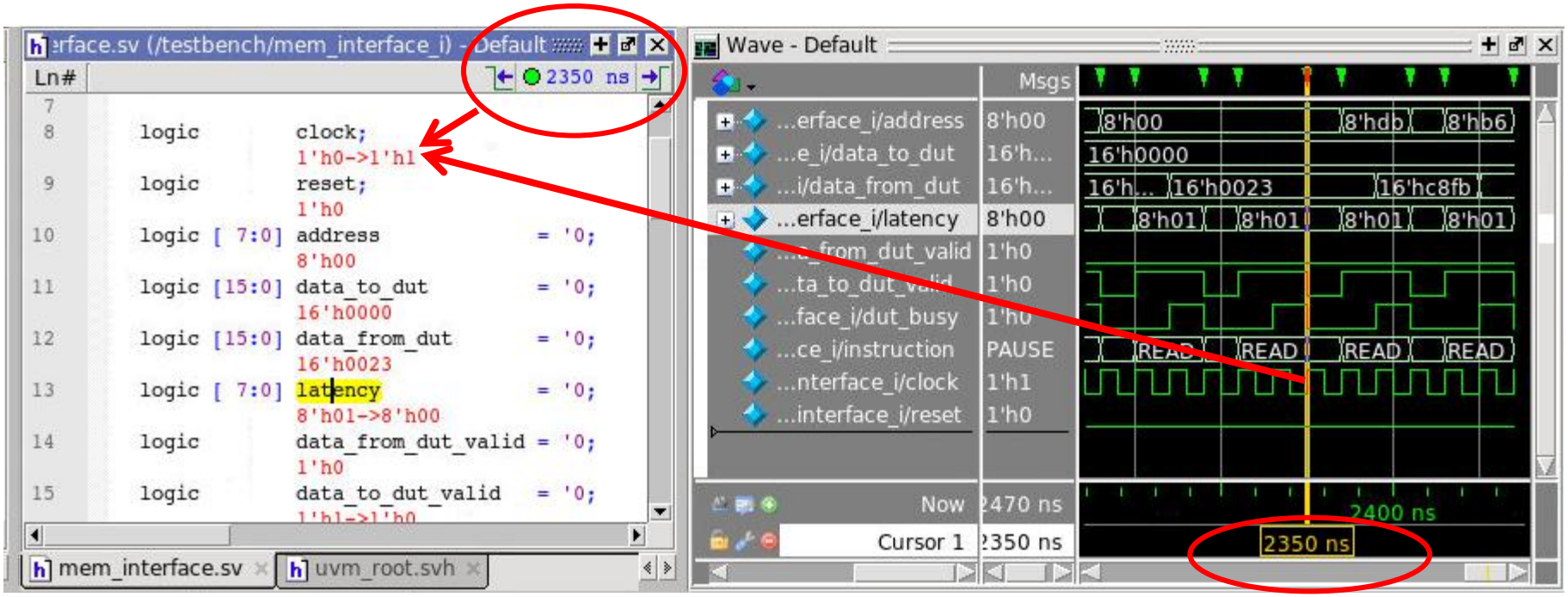
© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Cursor Linking



© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Cursors Based Navigation

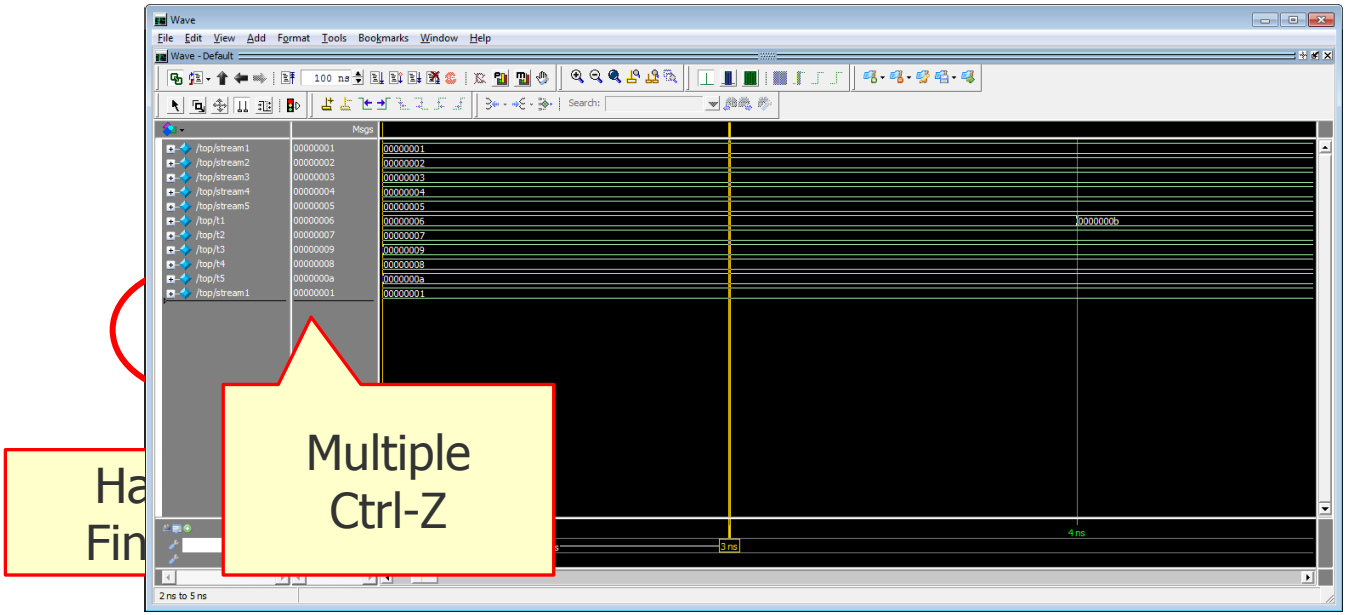


- Values tied to active cursor
  - Source, Object, Dataflow and Schematic windows
- Focus on events in time rather than which window to search in

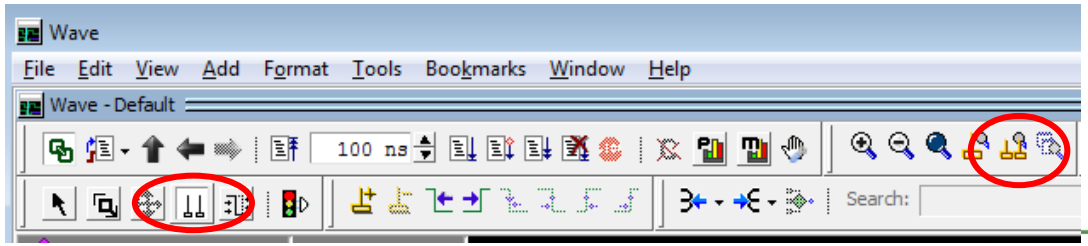
© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

New Wave Window Options

- Undo/redo of adding, moving, grouping or deleting signals
  - Ctrl-Z undo and Ctrl-Y redo

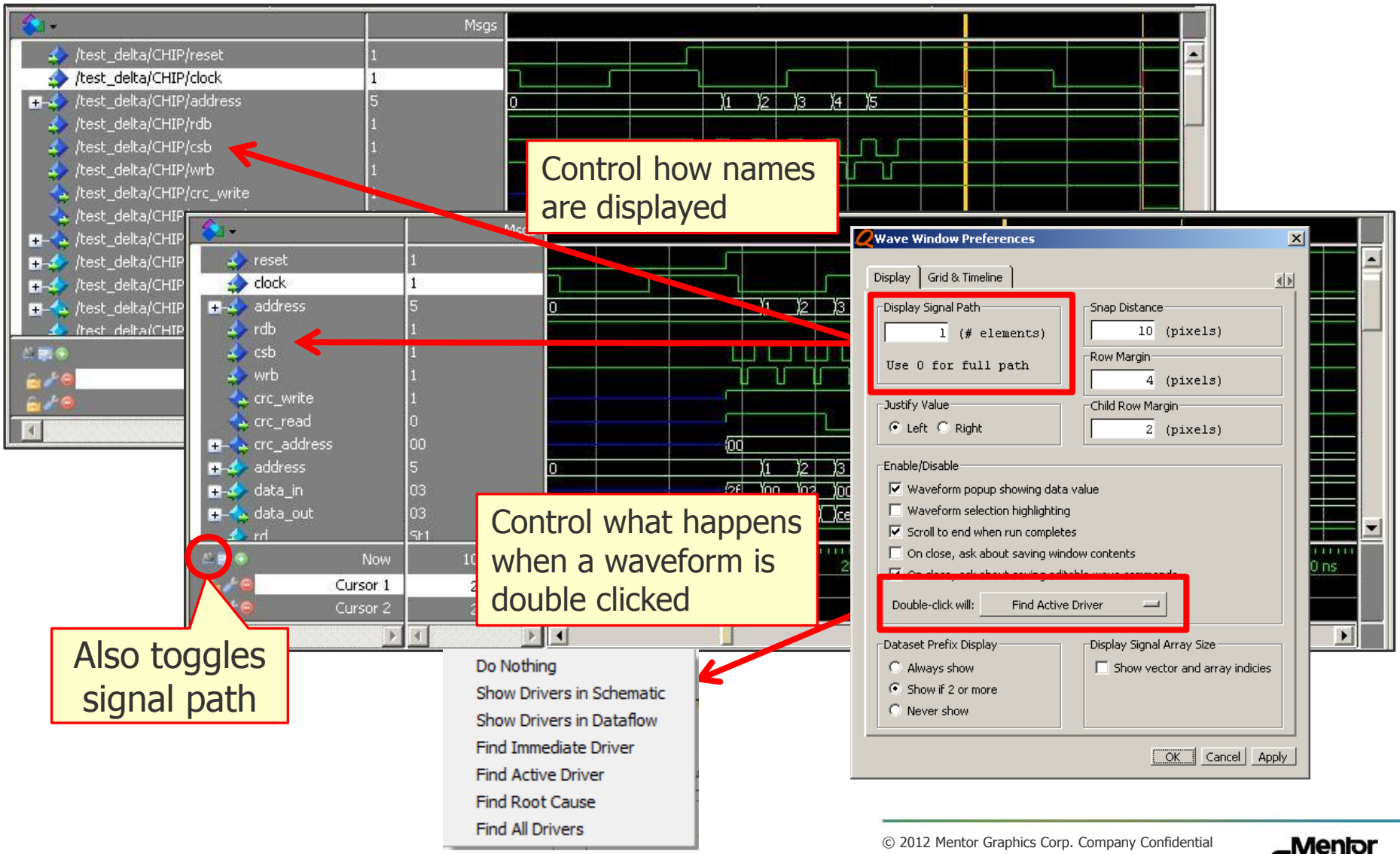


- Zoom Between Cursors



© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

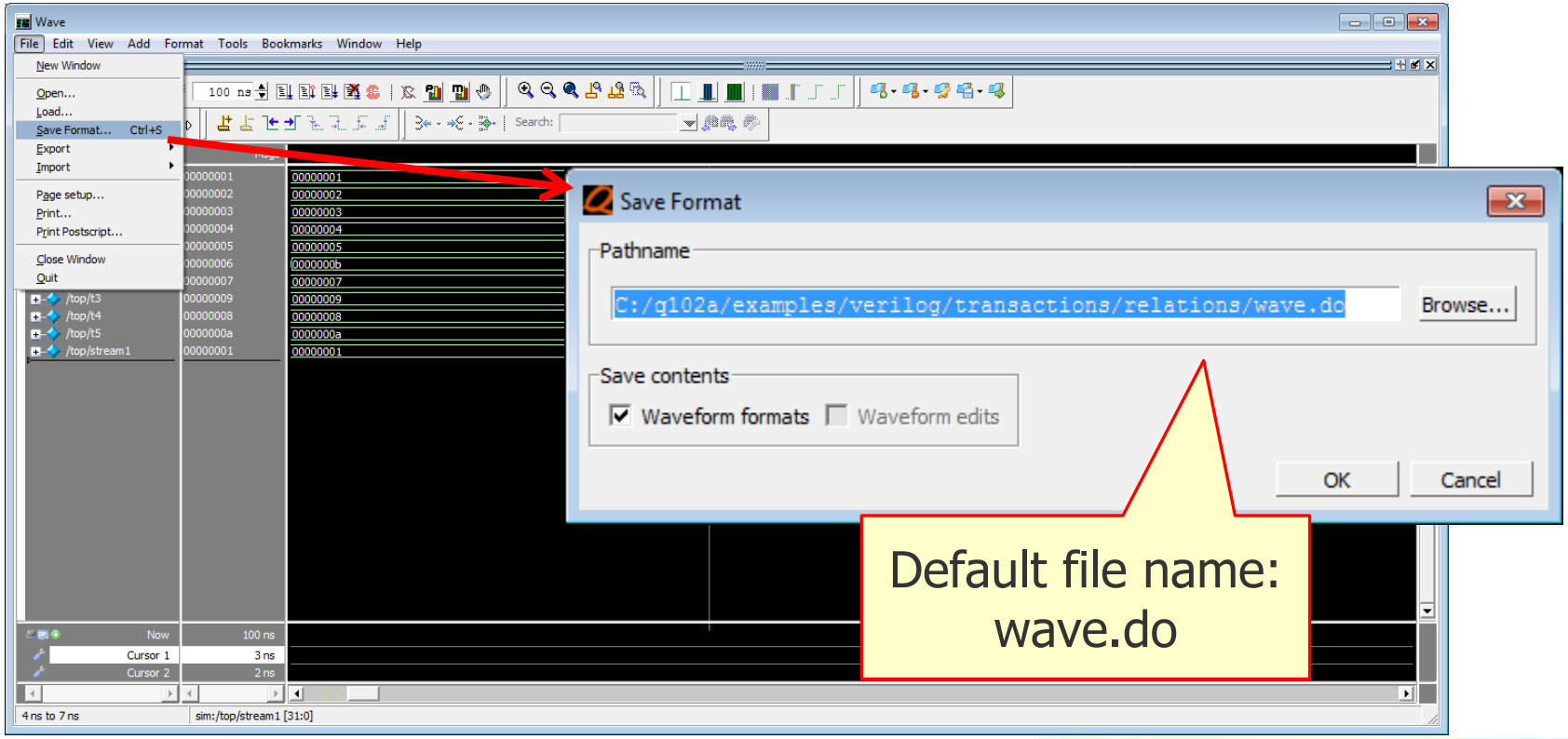
Display Properties



© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

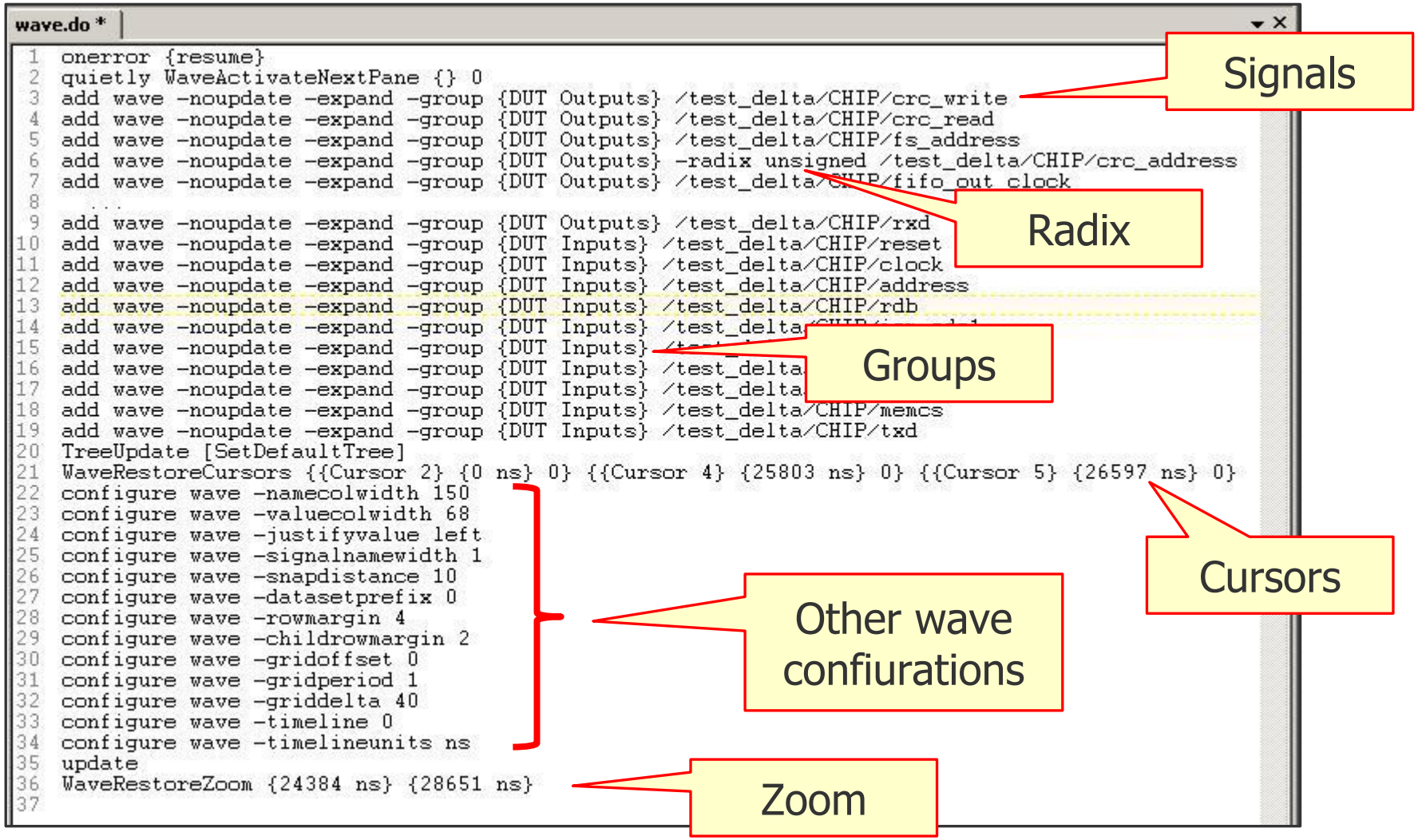
Saving the Wave Window Setup

- Once you have the wave window setup the way you like, you can save it for future reloading
  - Signals, zoom, cursors, radix, groups, dividers, etc.



© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

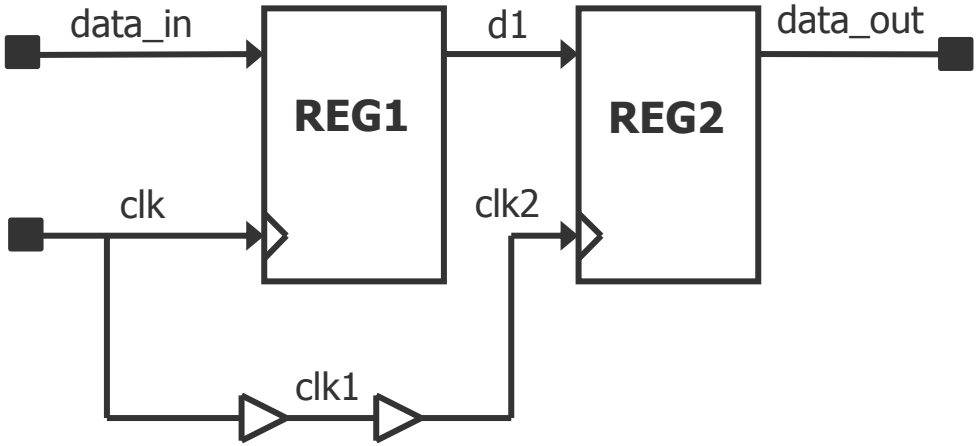
The "wave.do" File


wave.do *
1 onerror {resume}
2 quietly WaveActivateNextPane {} 0
3 add wave -noupdate -expand -group {DUT Outputs} /test_delta/CHIP/crc_write
4 add wave -noupdate -expand -group {DUT Outputs} /test_delta/CHIP/crc_read
5 add wave -noupdate -expand -group {DUT Outputs} /test_delta/CHIP/fs_address
6 add wave -noupdate -expand -group {DUT Outputs} -radix unsigned /test_delta/CHIP/crc_address
7 add wave -noupdate -expand -group {DUT Outputs} /test_delta/CHIP/fifo_out_clock
8 ...
9 add wave -noupdate -expand -group {DUT Outputs} /test_delta/CHIP/rxd
10 add wave -noupdate -expand -group {DUT Inputs} /test_delta/CHIP/reset
11 add wave -noupdate -expand -group {DUT Inputs} /test_delta/CHIP/clock
12 add wave -noupdate -expand -group {DUT Inputs} /test_delta/CHIP/address
13 add wave -noupdate -expand -group {DUT Inputs} /test_delta/CHIP/rdb
14 add wave -noupdate -expand -group {DUT Inputs}
15 add wave -noupdate -expand -group {DUT Inputs}
16 add wave -noupdate -expand -group {DUT Inputs}
17 add wave -noupdate -expand -group {DUT Inputs}
18 add wave -noupdate -expand -group {DUT Inputs} /test_delta/CHIP/memcs
19 add wave -noupdate -expand -group {DUT Inputs} /test_delta/CHIP/txd
20 TreeUpdate [SetDefaultTree]
21 WaveRestoreCursors {{Cursor 2} {0 ns} 0} {{Cursor 4} {25803 ns} 0} {{Cursor 5} {26597 ns} 0}
22 configure wave -namecolwidth 150
23 configure wave -valuecolwidth 68
24 configure wave -justifyvalue left
25 configure wave -signalnamewidth 1
26 configure wave -snapdistance 10
27 configure wave -datasetprefix 0
28 configure wave -rowmargin 4
29 configure wave -childrowmargin 2
30 configure wave -gridoffset 0
31 configure wave -gridperiod 1
32 configure wave -griddelta 40
33 configure wave -timeline 0
34 configure wave -timelineunits ns
35 update
36 WaveRestoreZoom {24384 ns} {28651 ns}
37
Signals
Radix
Groups
Cursors
Other wave confiurations
Zoom

© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Concurrency in Sequential Software

- While we are modeling concurrent HW, simulators must still execute events sequentially
  - Delta delays
- This creates unique debugging challenges

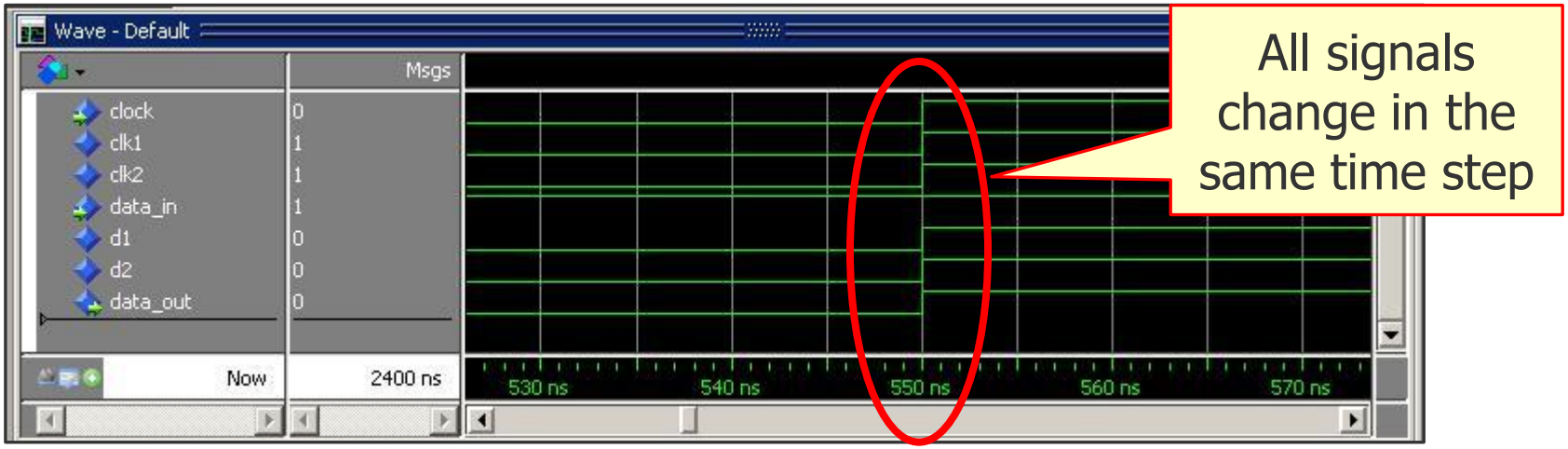



```
28      reg1: PROCESS (clock)
29      BEGIN
30          IF (clock'EVENT AND clock = '1') THEN
31              IF reset = '1' THEN
32                  d1 <= '0';
33              ELSE
34                  d1 <= data_in;
35              END IF;
36          END IF;
37      END PROCESS;
38
39      reg2: PROCESS (clock_2)
40      BEGIN
41          IF (clk2'EVENT AND clk2 = '1') THEN
42              IF reset = '1' THEN
43                  d2 <= '0';
44              ELSE
45                  d2 <= d1;
46              END IF;
47          END IF;
48      END PROCESS reg2;
49
50      clk1     <= clock;
51      clk2     <= clk1;
52      data_out <= d2;
```

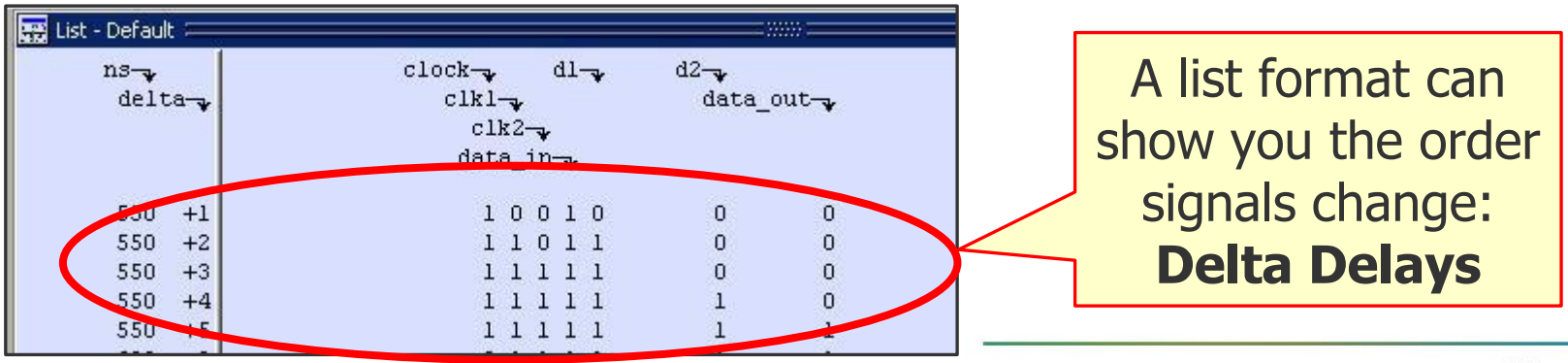

© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Traditional Viewing of Delta Delays

- Simultaneous events (delta delays) are not visible in typical waveform displays



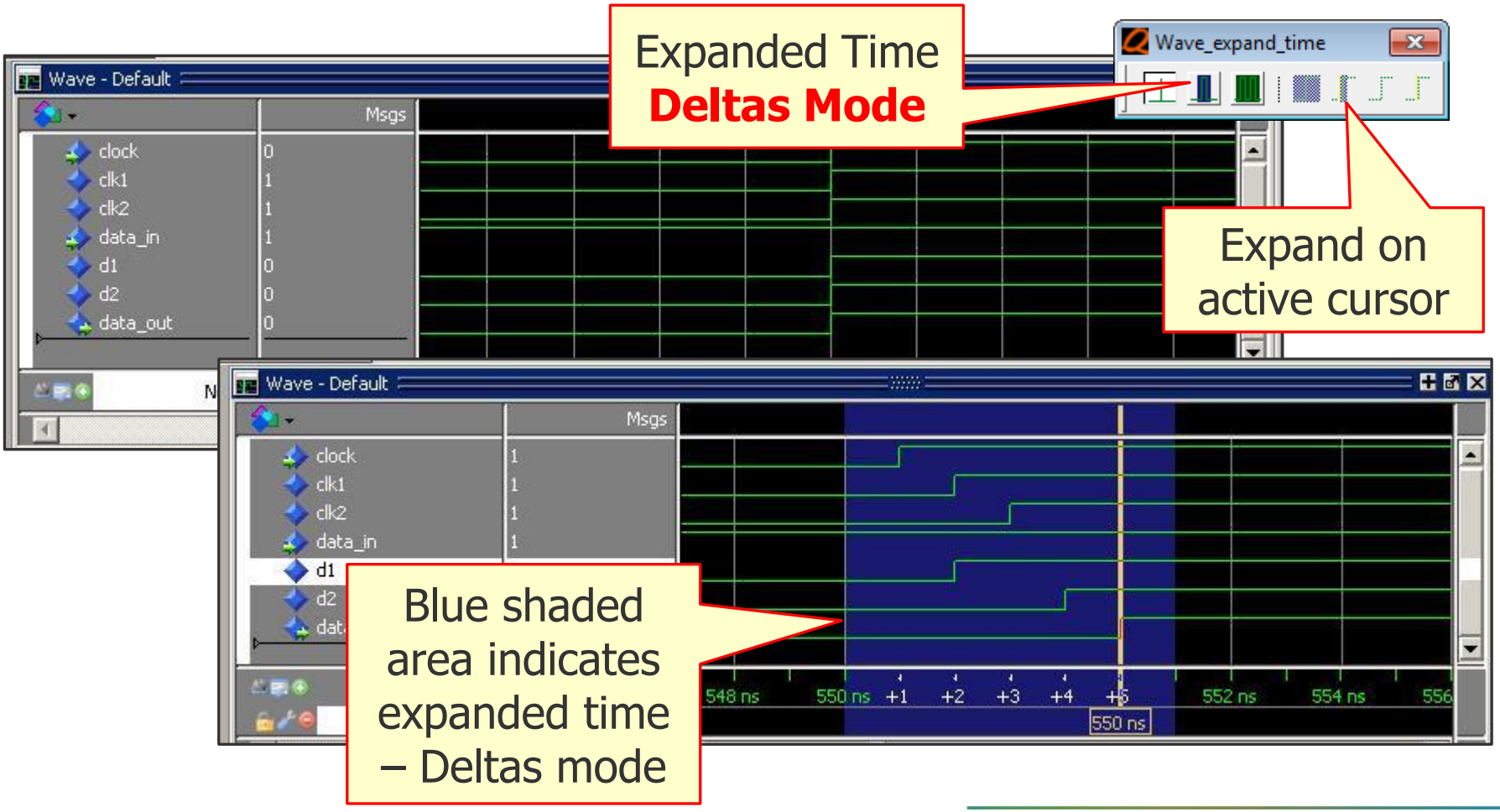
- Typically you need some sort of a list format


List - Default
ns
delta
clock
clk1
clk2
data_in
d1
d2
data_out
550 +1 1 0 0 1 0 0 0
550 +2 1 1 0 1 1 0 0
550 +3 1 1 1 1 1 0 0
550 +4 1 1 1 1 1 1 0
550 +5 1 1 1 1 1 1 1
A list format can show you the order signals change: **Delta Delays**

© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Deltas Viewing Mode

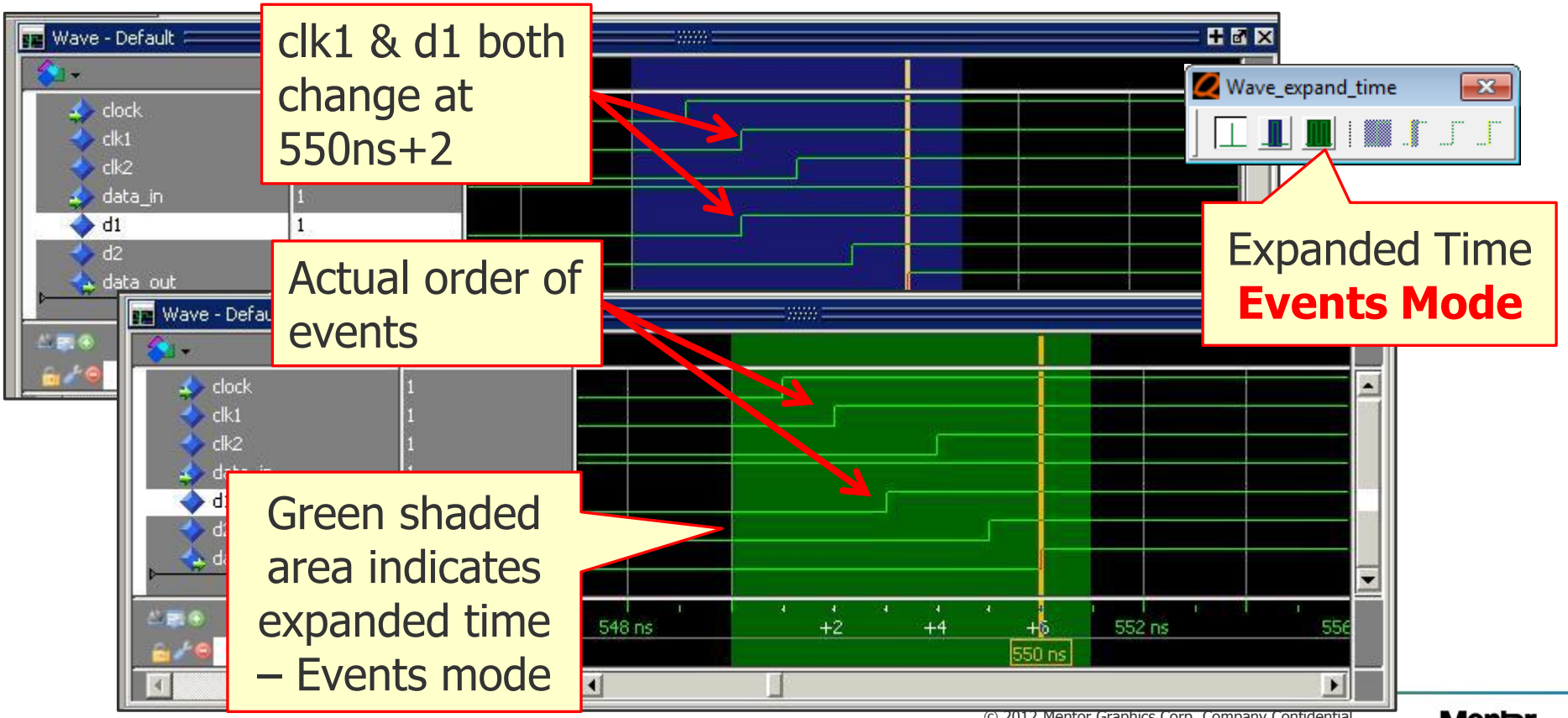
- Wave window can expand time to show delta delays



© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Events Viewing Mode

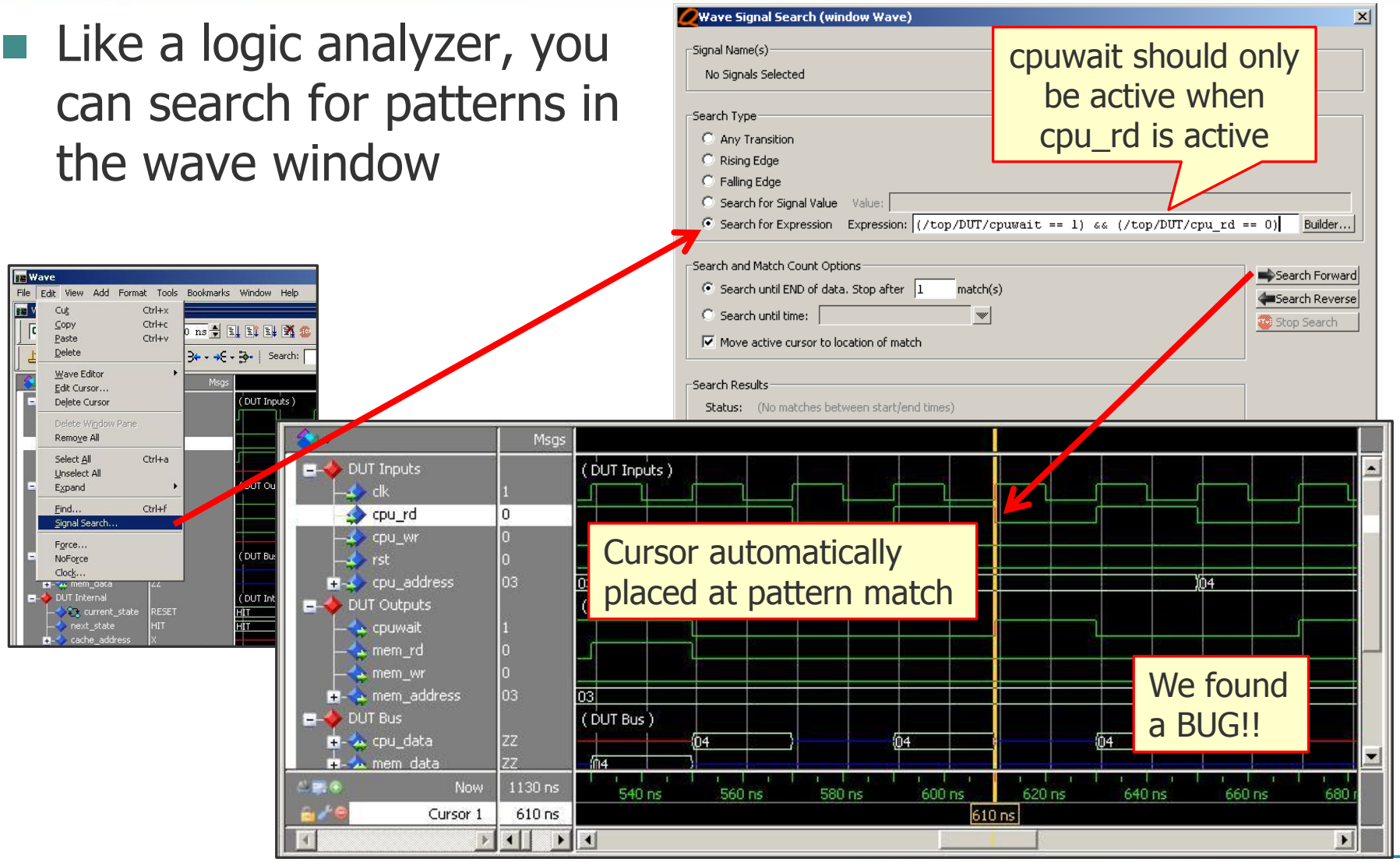
- The deltas mode can still have signals that change at the same delta
- Events mode does the next level of expansion



© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Searching for Signal Patterns

- Like a logic analyzer, you can search for patterns in the wave window

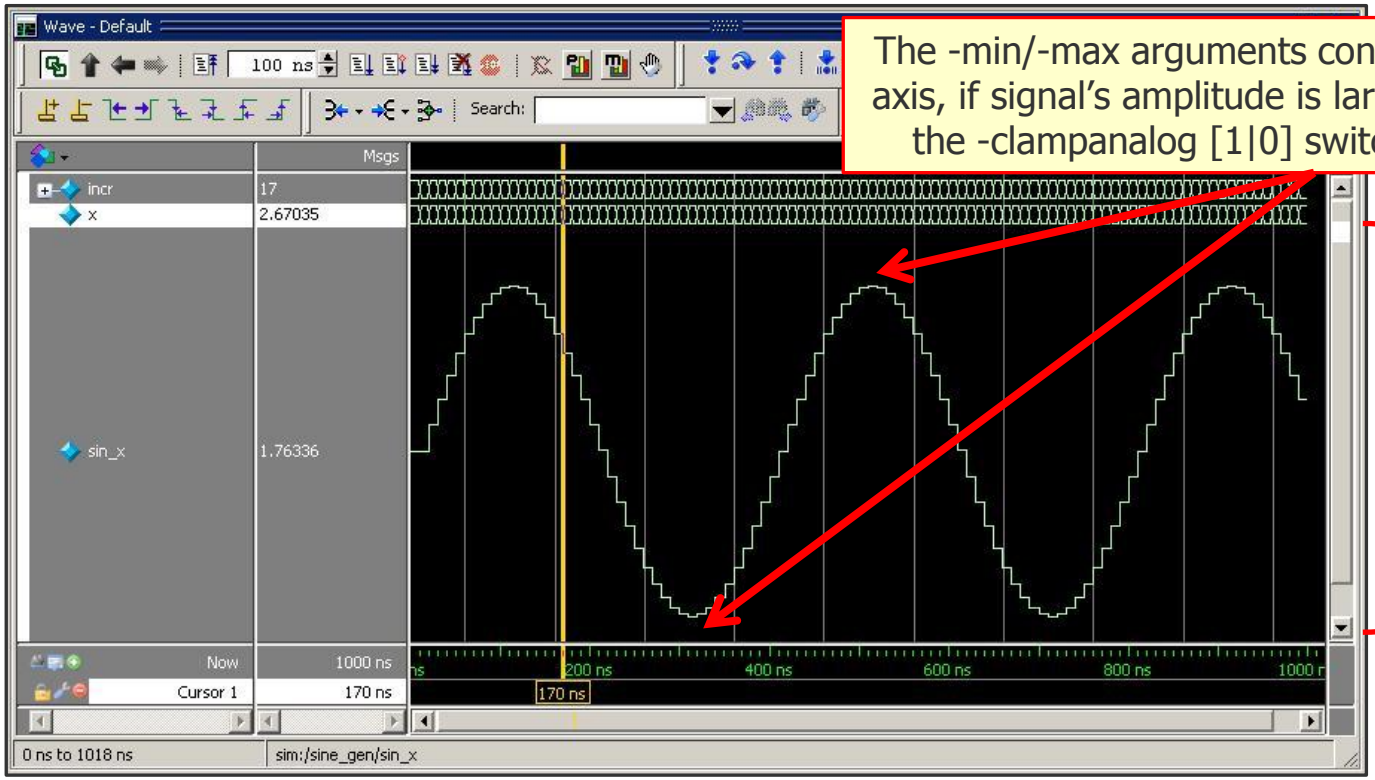


© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Displaying Signals with Analog Formatting

- The wave window has several options for viewing signals in an analog format

```
add wave -format Analog-Step -height 300 -max 4.0 -min -4.0 sin_x
```



The -min/-max arguments control the limits on the y-axis, if signal's amplitude is larger than the limits, use the -clampanalog [1|0] switch to control clipping

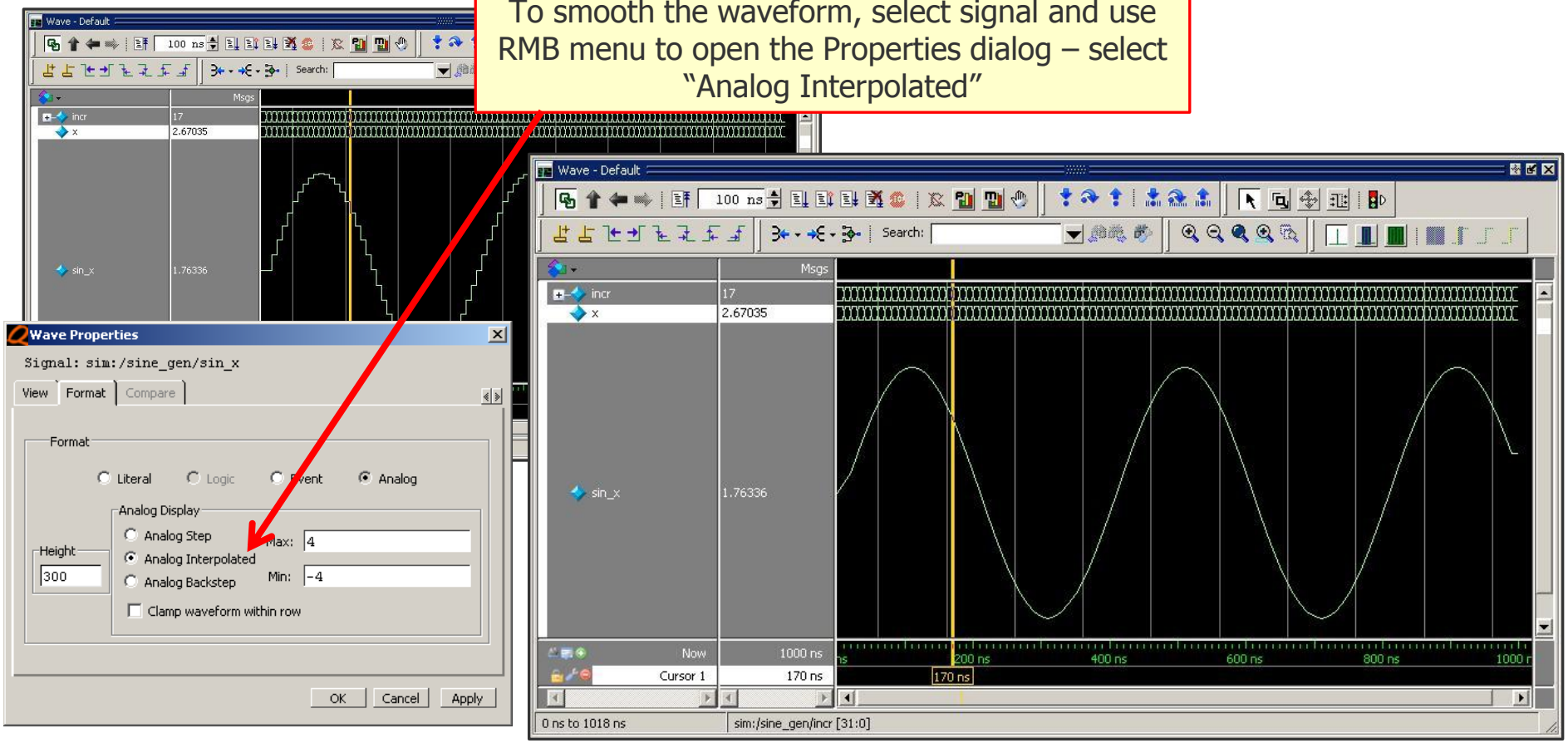
The -height argument controls drawing space in pixels, signals will automatically size themselves within this space

© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Smoothing Out the Waveform

- By default analog formatted signals are displayed using a step function

To smooth the waveform, select signal and use RMB menu to open the Properties dialog – select "Analog Interpolated"



© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

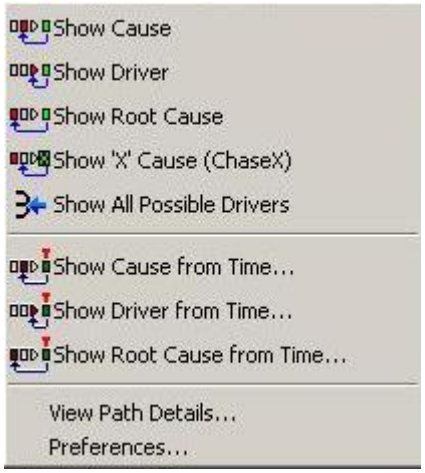
Agenda: Questa Debug Essentials

- General Questa Operations
- Managing Simulation Messages
- Source Level Debugging
- Viewing and Working with Waveforms
- Root Cause Analysis

© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Automatic Causality Tracing

- Two visual aids
  - Dataflow window
  - Schematic viewer
- Can be initiated from multiple debugging windows
  - Wave, Source, Objects, Schematic, Structure
- Find the cause of an unexpected output quickly
  - Tracing to the First Sequential Process (Show Cause)
  - Tracing to the Immediate Driving Process (Show Driver)
  - Tracing to the Root Cause (Show Root Cause)
  - Tracing to the Root Cause of an 'X' (Show 'X' Cause)
  - Finding All Possible Drivers (Show All Possible Drivers)
- Causality tracing flow

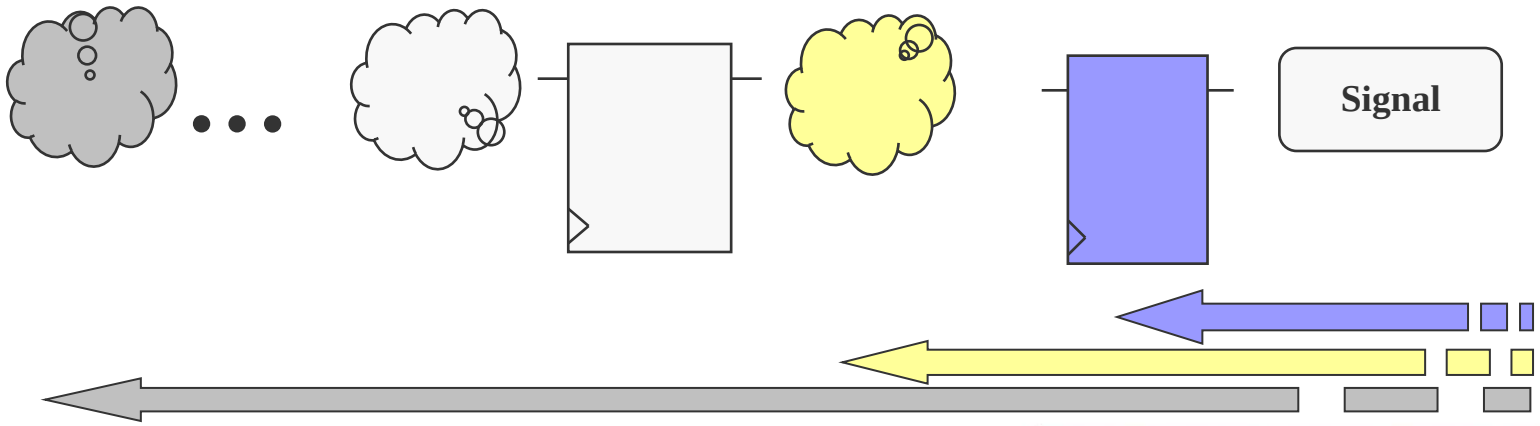



```
vlib work
vlog/vcom -f files.f
vopt -debugdb +acc top -o dbgver
vsim -debugdb dbgver
```


© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

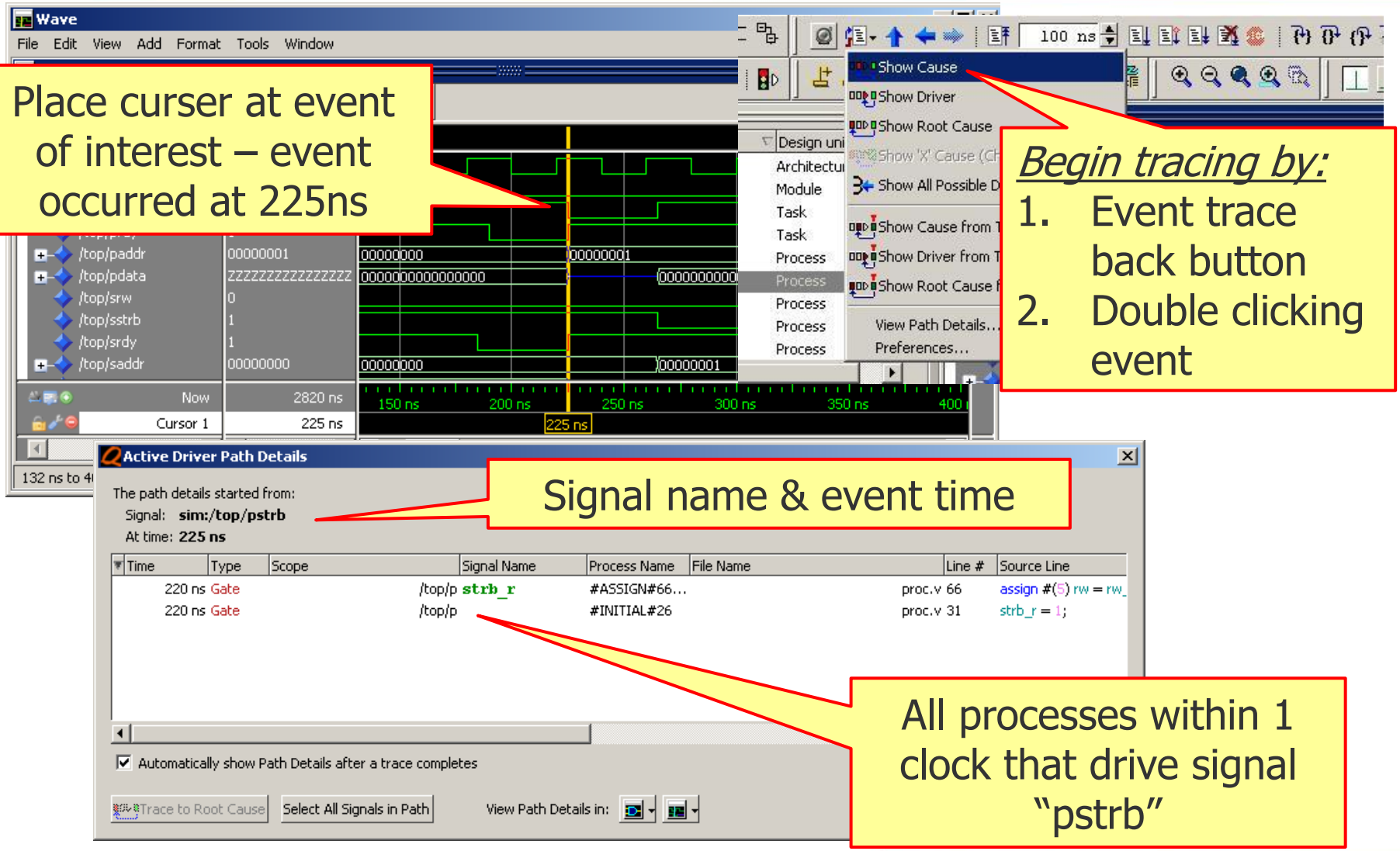
Causality Terminology

- Show Driver (vsim –novopt will activate only this option)
  - Highlight current active driving RTL
- Show Cause (needs vsim –voptargs=+acc)
  - Find cause of change one clock cycle back in time
- Show Root Cause (needs vsim -voptargs=+acc)
  - Perform multiple "show cause" automatically
- Show "X" Cause (needs vsim -voptargs=+acc)
  - Similar to "Show Root Cause" except for "X" only



© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Show Cause - Initiated from Wave Window



Place curser at event of interest – event occurred at 225ns

Begin tracing by:

1. Event trace back button
2. Double clicking event

Signal name & event time

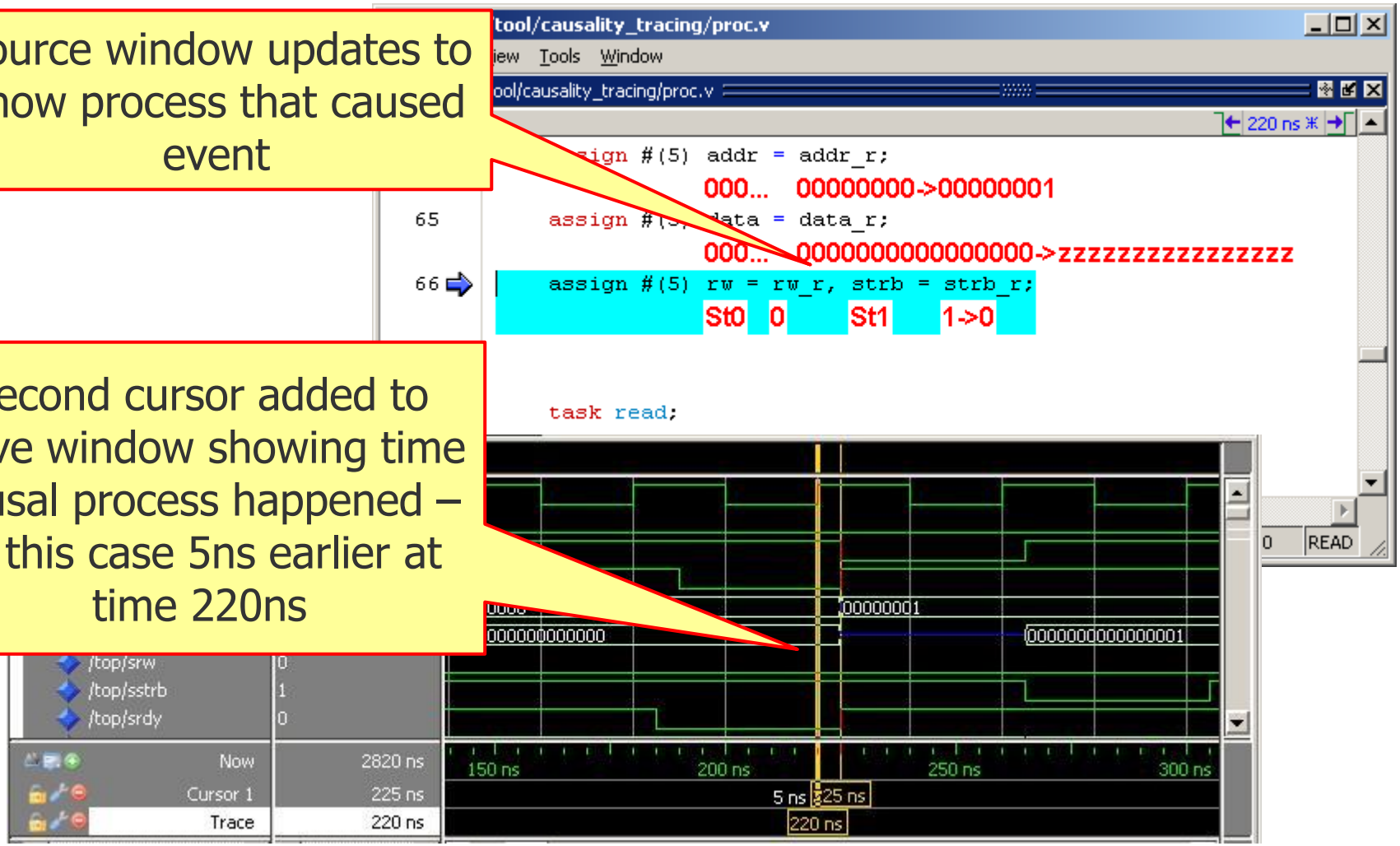
All processes within 1 clock that drive signal "pstrb"

© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Show Cause from Wave cont.

Source window updates to show process that caused event

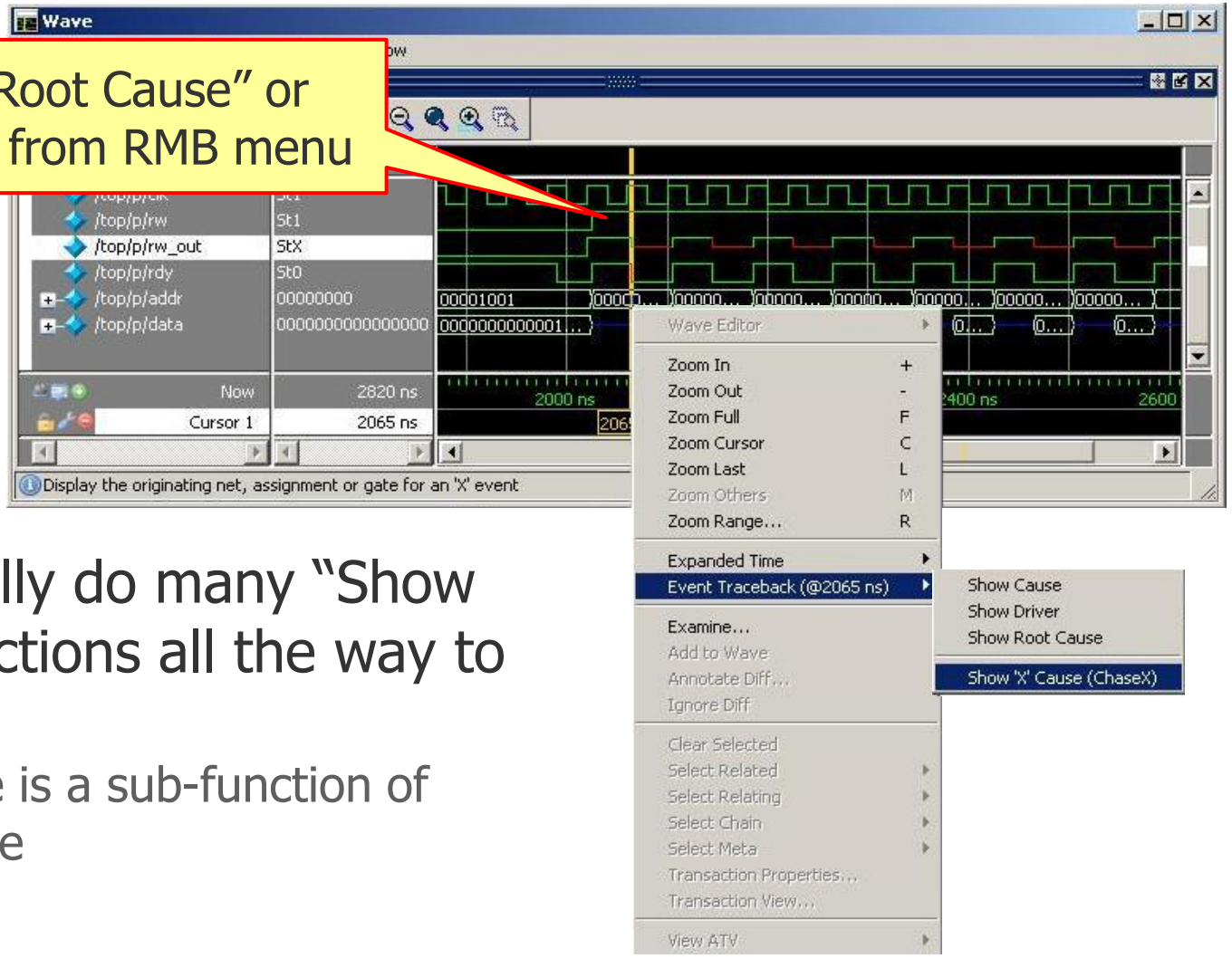
Second cursor added to wave window showing time causal process happened – in this case 5ns earlier at time 220ns



© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Show Root Cause & "X" Cause

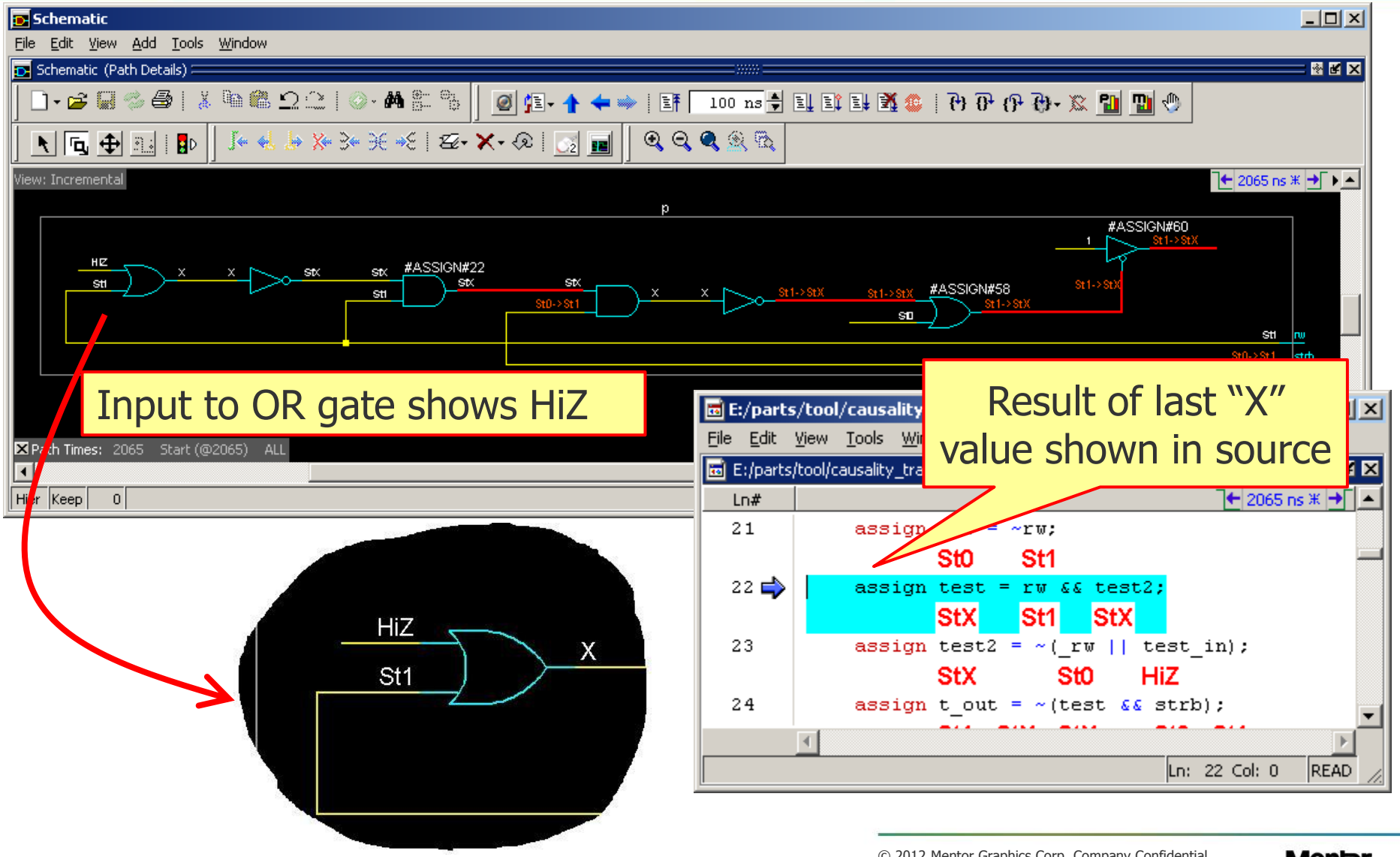
Select "Show Root Cause" or "Show X Cause" from RMB menu



- Automatically do many "Show Cause" functions all the way to the root
  - "X" cause is a sub-function of root cause

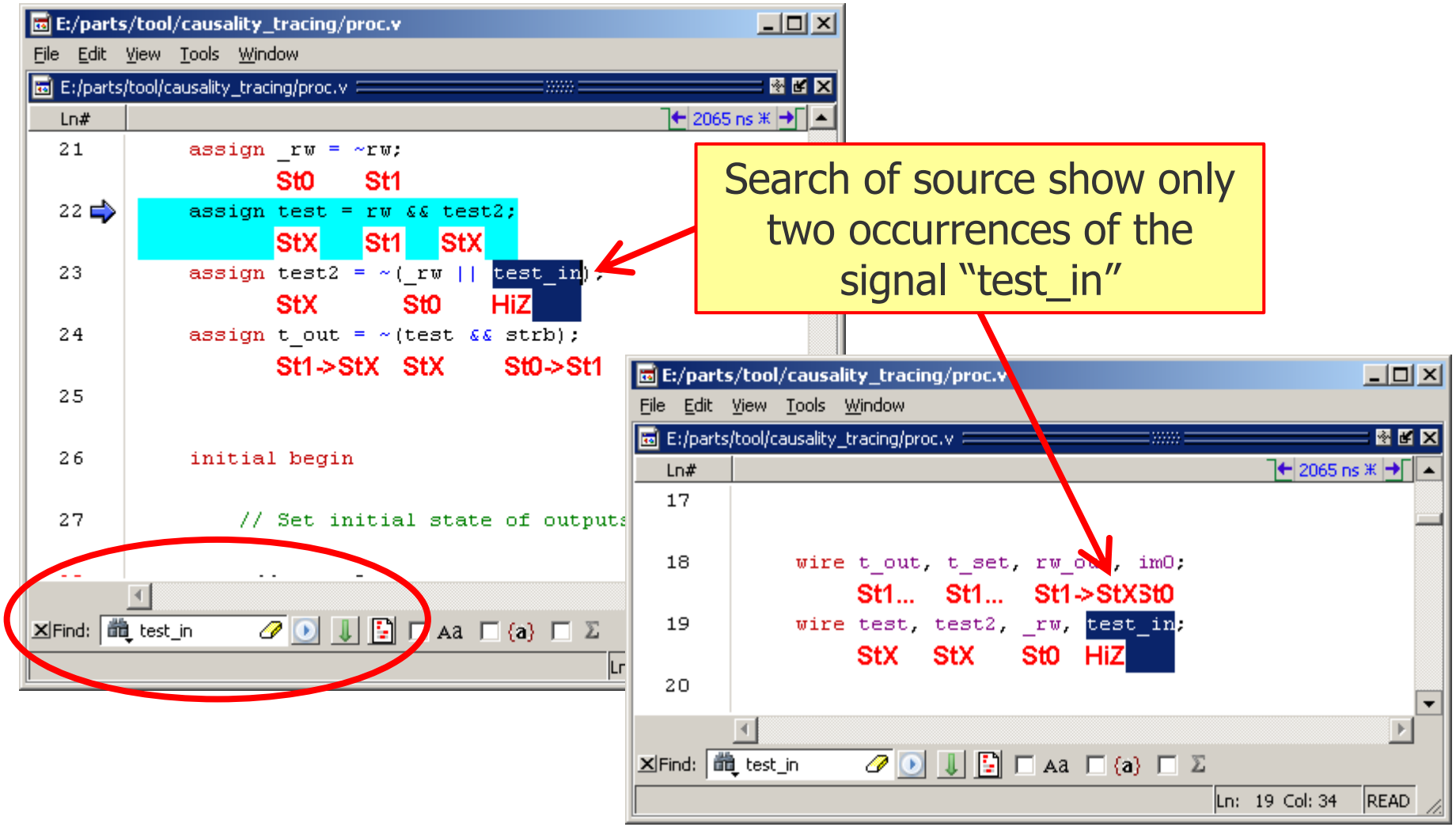
© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Show "X" Cause – Schematic Path



© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Searching the Source



© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Questa Update Summary

- Improved productivity for very day tasks
- Essential root cause analysis tools
- Foundation for adopting advanced verification processes

© 2012 Mentor Graphics Corp. Company Confidential
www.mentor.com

Mentor Graphics®

www.mentor.com